



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY**

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

**APLIKACE TECHNOLOGIE MOLECUBES V ROBOTICE**

MOLECUBES TECHNOLOGY APPLICATION IN ROBOTICS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Václav Vacek**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. Simeon Simeonov, CSc.**

**BRNO 2016**



# Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky  
Student: **Bc. Václav Vacek**  
Studijní program: Strojní inženýrství  
Studijní obor: Výrobní stroje, systémy a roboty  
Vedoucí práce: **doc. Ing. Simeon Simeonov, CSc.**  
Akademický rok: 2015/16

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Aplikace technologie MOLECUBES v robotice

### Stručná charakteristika problematiky úkolu:

Vytvoření fyzikálních modelů základních elementů. Realizace jednoduchých struktur za použití základních elementů

### Cíle diplomové práce:

1. Analýza problému
2. Vytvoření fyzikálních modelů základních elementů
3. Realizace jednoduchých struktur za použití základních elementů

### Seznam literatury:

KOLÍBAL, Zdeněk; KNOFLÍČEK, Radek. Morfologická analýza stavby průmyslových robotů. Vydání 1. Košice : Viena, 2000. 185 s. ISBN 80-88922-27-5.

EHRENBERGER, Zdeněk; KOLÍBAL, Zdeněk. Průmyslové roboty III : Robotické systémy vyšších generací. 1. vydání. Brno : VUT Brno, 1993. 145 s. ISBN 80-214-0530-0.

KOLÍBAL, Zdeněk. Průmyslové roboty I : Konstrukce průmyslových robotů a manipulátorů. 1. vydání. Brno : VUT Brno, 1993. 189 s. ISBN 80-214-0526-0.

SKAŘUPA, Jiří. Průmyslové roboty a manipulátory [online]. 1. vydání. Ostrava : Ediční středisko VŠB – TUO, 2007 [cit. 2010-04-17]. Dostupné z WWW: . ISBN 978-80-248-1522-0.

FESTO, [www.festo.com](http://www.festo.com), 1.12.2015

ABB, [www.abb.cz](http://www.abb.cz), 1.12.2015

FANUC, [www.fanucrobotics.cz](http://www.fanucrobotics.cz), 1.12.2015

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S.

---

doc. Ing. Petr Blecha, Ph.D.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **ABSTRAKT**

Cílem této práce je navrhnout a zrealizovat robota, který je tvořen ze stejných modulů. Tyto moduly se mohou samy spojovat, nebo odpojovat. Takto vznikají různé struktury robota. Zvolený problém je vyřešen konstrukčním návrhem modulu, který se může natáčet ve dvou osách a má připojovací konektory pro další moduly. Komunikace probíhá přes Wi-fi připojení s počítačem a úhly nutné pro rekonfiguraci jsou vypočítány inverzní kinematikou v programu Matlab. Na těchto modulech byl úspěšně proveden test rekonfigurace.

## **ABSTRACT**

The aim of the thesis is to propose and make a robot, which is made of identical modules. These modules are able to connect or disconnect themselves and thanks to this feature new structures of robot can be achieved. This problem is solved by the design proposal of a module, which is capable to rotate in two axis and has connection connectors for other modules. Communication is carried out by Wi-fi connection to the computer and angles required for reconfiguration are calculated by inverse kinematics in Matlab program. On these modules the reconfiguration test was succesfully demonstrated.

## **KLÍČOVÁ SLOVA**

Molecubes, modulární roboty, sebekonfigurující se roboty, přímá kinematika, inverzní kinematika, mikrokontrolér

## **KEYWORDS**

Molecubes, modular robots, self-reconfigurable robots, forward kinematics, inverse kinematics, microcontroller



## **BIBLIOGRAFICKÁ CITACE**

VACEK, V. *Aplikace technologie MOLECUBES v robotice*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 81 s. Vedoucí diplomové práce doc. Ing. Simeon Simeonov, CSc.





## **PODĚKOVÁNÍ**

Tímto bych chtěl poděkovat svému vedoucímu diplomové práce doc. Ing. Simeonu Simeonovi, CSc. za odborné vedení diplomové práce. Dále bych chtěl poděkovat své snoubence Verunce a své rodině za podporu během studia a psaní diplomové práce.



## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením doc. Ing. Simeona Simeonova, CSc. a s použitím literatury uvedené v seznamu.

V Brně dne 25.5.2016

.....

Václav Vacek



# OBSAH

<b>1</b>	<b>ÚVOD .....</b>	<b>15</b>
<b>2</b>	<b>MODULÁRNÍ SEBEKONFIGUROVATELNÉ ROBOTY .....</b>	<b>17</b>
2.1	Modulární sebekonfigurovatelné roboty .....	17
2.2	Výhody modulárních sebekonfigurovatelných robotů .....	17
2.3	Aplikace modulárních sebekonfigurovatelných robotů .....	17
2.4	Architektura uspořádání modulů .....	18
2.4.1	Mřížové uspořádání .....	18
2.4.2	Řetězová struktura .....	18
2.4.3	Mobilní roboty .....	19
2.4.4	Hybridní struktura .....	19
2.5	Rekonfigurace .....	19
2.5.1	Deterministická .....	19
2.5.2	Stochastická .....	19
2.6	Design .....	20
2.6.1	Homogenní .....	20
2.6.2	Heterogenní .....	20
2.7	Architektura řízení .....	20
2.7.1	Centralizované řízení .....	20
2.7.2	Decentralizované řízení .....	21
2.7.3	Komunikace mezi moduly .....	21
2.8	Příklady robotů .....	21
2.8.1	M-TRAN III .....	21
2.8.2	Polybot G3 .....	23
2.8.3	ATRON .....	23
<b>3</b>	<b>KONSTRUKČNÍ NÁVRH .....</b>	<b>25</b>
3.1	Požadavky na konstrukční návrh .....	25
3.2	Volba struktury .....	25
3.3	Spojení dvou modulů .....	26
3.4	Propojení modulů .....	28
3.5	Servomotor .....	29
3.6	Výroba modulu .....	31
3.7	Kontrola spojení modulů .....	32
3.8	Uchycení vývojové desky .....	32
3.9	Průchod kabelu mezi horní a dolní částí modulu .....	33
3.10	Použité matice .....	34
3.11	Uložení baterií .....	34
<b>4</b>	<b>VOLBA MIKROKONTROLÉRU A NAPÁJENÍ .....</b>	<b>35</b>
4.1	Volba mikrokontroléru .....	35
4.1.1	Mikrokontrolér Arduino .....	35
4.1.2	Vývojová deska NodeMCU .....	36
4.2	Napájení robota .....	37
4.2.1	Požadované napětí a proud .....	37
4.2.2	Návrh baterií .....	38
4.2.3	Držák baterie .....	39
4.2.4	Změna velikosti napětí .....	39
4.2.5	Napájení modulu z externího zdroje .....	39

<b>5</b>	<b>PROGRAMOVÁNÍ MIKROKONTROLÉRU .....</b>	<b>41</b>
5.1	Vývojové prostředí .....	41
5.2	Připojení NodeMCU .....	41
5.3	Terminologie používaná v sítích .....	42
5.4	Možnosti připojení k Wi-fi modulu .....	42
5.5	Ovládání servomotorů .....	43
5.6	Program pro Molecube – server .....	44
5.7	Program pro Molecube – přístupový bod .....	47
<b>6</b>	<b>PŘÍMÁ A INVERZNÍ ÚLOHA KINEMATIKY .....</b>	<b>49</b>
6.1	Využití přímé a inverzní kinematiky pro rekonfiguraci robota .....	49
6.2	Přímá úloha kinematiky .....	49
6.2.1	Dvě postupné rotace .....	51
6.2.2	Transformační matice .....	51
6.3	Inverzní úloha kinematiky .....	51
6.3.1	Pseudoinverze .....	52
6.3.2	Transpozice .....	52
6.3.3	Metoda tlumených nejmenších čtverců .....	52
6.4	Aplikace numerické metody .....	53
6.5	Využití parametru lambda .....	53
<b>7</b>	<b>REKONFIGURACE ROBOTA .....</b>	<b>55</b>
7.1	Průběh rekonfigurace .....	55
7.2	Požadavky na rekonfiguraci .....	55
7.3	Vstup a výstup řešení .....	55
7.4	Odhad řešení .....	56
7.5	Funkce pro výpočet natočení .....	56
7.6	Funkce pro dopřednou kinematiku .....	57
7.7	Funkce pro Jakobiho matici .....	58
7.8	Omezení rozsahu pohybu v kloubu .....	58
7.9	Urychlení konvergence pro triviální případy .....	60
7.10	Orientace koncového efektoru .....	61
7.11	Počáteční podmínky .....	62
7.12	Grafické uživatelské rozhraní .....	63
7.13	Odeslání http příkazu z Matlabu .....	64
7.14	Testování rekonfigurace .....	64
7.14.1	Podmínky testování .....	64
7.14.2	Testování propojení v horizontální rovině .....	64
7.14.3	Testování propojení ve vertikální rovině .....	66
7.15	Poznatky z průběhu rekonfigurace .....	67
7.16	Návrhy na zlepšení .....	69
7.17	Manipulace ve 3D prostoru .....	69
7.18	Vlastnosti a využití robota Molecube .....	70
<b>8</b>	<b>ZÁVĚR.....</b>	<b>71</b>
<b>9</b>	<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>73</b>
<b>10</b>	<b>SEZNAM OBRÁZKŮ A SYMBOLŮ .....</b>	<b>77</b>
10.1	Seznam obrázků .....	77
10.2	Seznam symbolů .....	79
<b>11</b>	<b>SEZNAM PŘÍLOH.....</b>	<b>81</b>

# 1 ÚVOD

Technologie Molecubes se zabývá modulárními sebekonfiguračními roboty. Tyto roboty jsou tvořeny mnoha stejnými moduly. Každý modul má pouze jednoduché schopnosti, ale různými způsoby vzájemného propojení moduly dokáží vytvořit rozmanité struktury a plnit tak různé úkoly. Sebekonfigurace zajišťuje schopnost změny struktury robota na jinou strukturu.

Cílem této práce je zjistit, jakým způsobem mohou být tyto roboty realizovány a popsat několik konkrétních řešení. Dále navrhnout vlastní konstrukční řešení, navrhnout řízení a napájení modulů a tento návrh zrealizovat. Nezbytnou součástí je vypočítávání úhlů natočení pro jednotlivé moduly tak, aby mohla proběhnout rekonfigurace. Na závěr provést test, kdy robot změní sám svoji strukturu pomocí zrealizovaných modulů.





## 2 MODULÁRNÍ SEBEKONFIGUROVATELNÉ ROBOTY

### 2.1 Modulární sebekonfigurovatelné roboty

Modulární sebekonfigurovatelné roboty se skládají z velkého množství stejných modulů, které díky různému vzájemnému propojení mohou vytvářet rozmanité struktury. Tvar struktury je pak takový, aby co nejlépe vyhovoval zadanému úkolu. Může se jednat o prolézání dírou, přesunování se odvalováním modulů nebo o vytvoření složité struktury s několika rameny. [1]

Nejznámějším příkladem mobilních sebekonfiguračních robotů je fiktivní tekutinkový robot z filmu od Jamese Camerona Terminator 2: Den Zúčtování. V tomto filmu futuristický robot z miliónů mikroskopických modulů může měnit svůj tvar a reorganizovat se pro vykonání různých úkolů. [1]

### 2.2 Výhody modulárních sebekonfigurovatelných robotů

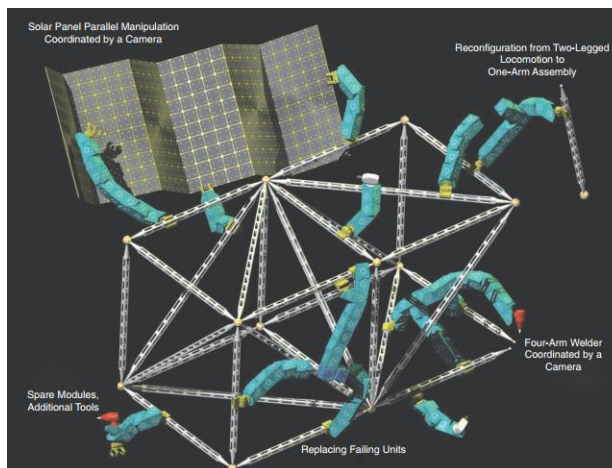
Sebekonfigurovatelné roboty mají několik unikátních znaků, které je činí zajímavými z inženýrského pohledu: [2]

- Robustnost a odolnost (Moduly mají vysoký počet stupňů volnosti. Pokud dojde k poškození jednoho modulu, neselže celé zařízení, ale poškozené moduly se mohou nahradit jinými moduly.)
- Univerzálnost (Vyjadřuje schopnost modulů vykonat různé druhy zadaných úkolů.)
- Schopnost adaptace (Možnost měnit svůj tvar plynule podle změny prostředí nebo při změně požadovaného úkolu.)
- Nízká cena v porovnání s jejich složitostí (Jeden modul je poměrně drahý, ale díky sériové produkci stejných modulů se náklady sníží.)

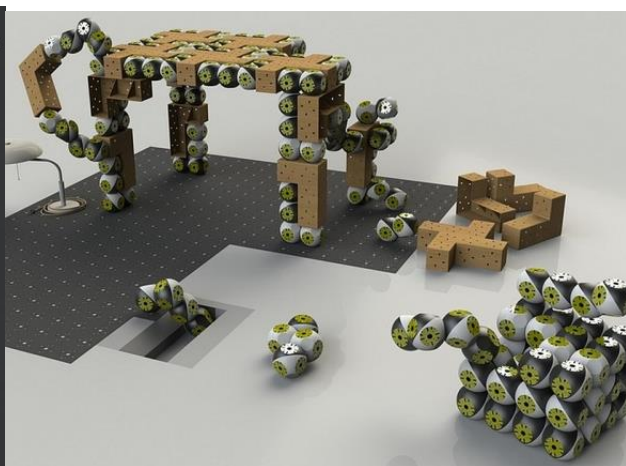
### 2.3 Aplikace modulárních sebekonfigurovatelných robotů

Využití těchto robotů je vhodné především tam, kde požadujeme plnění různých úkolů a kde předem neznáme operační podmínky. Příklady využití: [1][14]

- Využití ve vesmíru (Malé rozměry jednotlivých modulů jsou vhodné k jednoduchému převozu. Pokud se některý modul během mise zničí, nemusí být mise neúspěšná, neboť může být nahrazen jiným modulem (Obr. 1).)
- Hledání osob po zemětřeseních (v předem neznámých podmínkách se robot díky vhodné struktuře dostane k ohrožené osobě, přivolá pomoc a může vybudovat ochranný štít pro ohroženou osobu)
- Hašení požárů, průzkum bojiště, podmořská těžba, stavba velkých objektů
- Zábavní průmysl
- Servisní roboty (např. vize „Bucket of Stuff“ (kbelík věcí) kdy koncový uživatel má k dispozici velké množství modulů, které za něj dělají různé domácí práce, nebo Roombot – firma s robotickým nábytkem (Obr. 2))



Obr. 1 Využití ve vesmíru [24]



Obr. 2 Robotický nábytek Roombot [15]

## 2.4 Architektura uspořádání modulů

### 2.4.1 Mřížové uspořádání

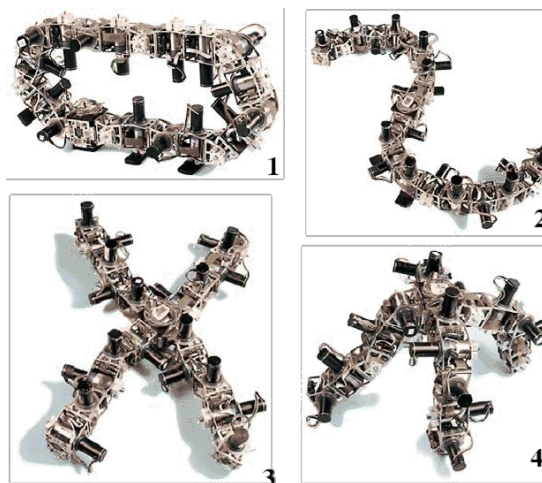
Jedná se o nominální uspořádání do 2D nebo 3D mřížové struktury. Pro tuto kategorii jsou jednotlivé pozice, které může modul obsadit, určeny diskrétně. V kontrastu s řetězovým uspořádáním, kde se moduly volně pohybují kontinuálně v prostoru, toto uspořádání značně zjednodušuje rekonfigurační proces. Kinematika a detekce kolizí je poměrně jednoduchá ve srovnání s ostatními uspořádáními. Příkladem je robot Miche (Obr. 3). [1]

### 2.4.2 Řetězová struktura

V řetězové struktuře jsou moduly spojeny sériově za sebou. Mohou tak vytvářet stromové a smyčkové struktury. Tyto moduly jsou pak umístěny v prostoru, takže koordinace a rekonfigurace je poměrně komplexní. Při větším počtu modulů jsou již komplikované výpočty pro dopřednou a inverzní kinematiku, plánování pohybů a detekci kolizí. Příkladem je robot Polybot (Obr. 4). [1]



Obr. 3 Robot Miche [6]



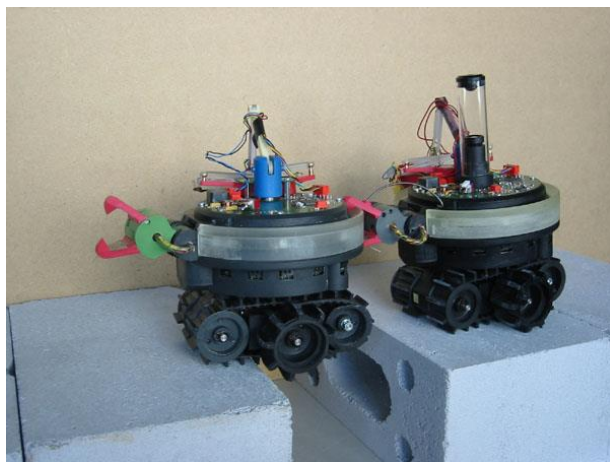
Obr. 4 Polybot [7]

### 2.4.3 Mobilní roboty

Jsou tvořeny moduly, které se mohou pohybovat i pokud jsou rozpojené od ostatních modulů. Když se spojí, vytvoří mřížovou strukturu nebo řetězovou strukturu. Příkladem těchto modulů jsou moduly s koly, které umožňují pohyb modulů a následné spojení modulů. Dalším příkladem jsou moduly, které se vznášejí v kapalině nebo ve vesmíru a spojují se s ostatními moduly. Příkladem mobilního robota je robot Swarm-bot. Je tvořen ze samostatných dílů, které se mohou propojit pro překonání větších překážek (Obr. 5). [1]

### 2.4.4 Hybridní struktura

Kombinuje možnosti řetězové struktury a mřížové struktury. Díky tomu je možné dosáhnout vysoké obratnosti a rekonfigurovatelnosti. Příkladem hybridní struktury je robot M-TRAN III (Obr. 6). [1]



Obr. 5 Swarm-bot mobilní robot [8]



Obr. 6 M-TRAN III [1]

## 2.5 Rekonfigurace

### 2.5.1 Deterministická

V deterministických systémech se moduly pohybují nebo jsou manipulovány přímo z jedné pozice do nové pozice v řetězové nebo mřížové struktuře. Pozice každého modulu je známá po celou dobu. Lze spočítat, kolik času zabere systému změna z jedné konfigurace na jinou. Modulový rekonfigurační mechanismus požaduje kontrolní strukturu, která umožní koordinovat a vykonávat rekonfigurační úkoly. [1]

### 2.5.2 Stochastická

Ve stochastickém systému se moduly pohybují ve 2D nebo 3D prostoru náhodně a vytvářejí různé struktury. Když některý modul narazí na jiný modul, rozhodne se, jestli se připojí nebo nepřipojí. Čas, za který moduly vytvoří požadovanou strukturu, se dá vypočítat pravděpodobnostně. [1]

## 2.6 Design

### 2.6.1 Homogenní

Je modulární robotický systém, který tvoří stejné moduly. Výhoda spočívá v tom, že přidáváním stejných modulů rostou možnosti a úkoly, které je systém schopný splnit. Nevýhodou je, že je modulů obvykle potřeba větší množství, než pro heterogenní moduly. [5]

### 2.6.2 Heterogenní

Modulární systém je tvořen různými moduly, kde každý dokáže plnit nějakou specifickou funkci. Jeho výhodou jsou kompaktnější rozměry díky použití modulů, které jsou pro splnění úkolu potřeba. Nevýhodou je potom vzrůst složitosti designu, výroby a simulačních metod. [5]

## 2.7 Architektura řízení

Každý modul má pouze jednoduché schopnosti a proto řízení jednoho modulu je snadné. Řízení mnoha modulů se však stává složitým problémem. [1]

### 2.7.1 Centralizované řízení

Každý modul má vlastní mikrokontrolér, kterým ovládá svůj pohon. Hlavní jednotka komunikuje potom s moduly a požaduje určitá natočení motorů. Funkčnost systému je tedy následující - určená jednotka odesílá příkazy všem individuálním modulům a synchronizuje akci celého systému. [1]

Zajímavou možností rozšíření přímého řízení je implementovat metodu založenou na biologickém hormonovém systému. Základní myšlenkou je, že tato zpráva se může šířit mezi moduly na nižší úrovni než celkové řízení. Obvyklá biologická analogie se objevuje, když se člověk například dotkne něčeho horkého, nebo když je ve studeném prostředí (rychlé odtržení ruky, změna struktury kůže). Toto zjednodušuje celkové řízení a monitoring celého systému. [1]

Existují tři možnosti řízení:

#### 1) Časová tabulka

Jednoduchá metoda implementace této metody je použití kontrolní tabulky. Ta se skládá z  $n \times m$  matice kde  $m$  je počet modulů a  $n$  je číslo kroků. V každé buňce v tabulce je uložen požadovaný úhel pro jednotlivý modul. Kontrolér prochází skrz tabulku řádek po řádku a odesílá tyto hodnoty úhlů korespondujícím modulům. Běžně je procházení tabulkou specifickou rychlostí, takže vertikální osa může reprezentovat čas. Čas mezi kroky je určen rychlostí motorů v kloubech modulu. [1] [9]

#### 2) Metoda posílané zprávy

Technologie je modelovaná stejným způsobem, jako jediný hormon dokáže vytvořit různé reakce v těle. Nemá poslána specifická instrukce každému modulu, ale jednoduchá zpráva prochází skrz jeden modul do dalšího. Některé moduly si zprávu modifikují, a proto vyvolá odlišné reakce modulů. Například stejná zpráva může změnit úhel motoru jednoho motoru, nezměnit nic u dalšího motoru a sama se smazat ve třetím modulu. [9]

#### 3) Hierarchická organizace

Tato metoda se hodí zvláště pro řetězové modulové roboty. Jedná se o rozdělení robota na hierarchické části jako prst, ruka, rameno (podobně jako hierarchie těla). Díky této organizaci se zjednoduší programování, protože můžeme řídit každou část zvlášť. [9]

### 2.7.2 Decentralizované řízení

Využívá zpravidla lokální řízení. Decentralizované řízení používá pravidla, která jsou stejná pro všechny moduly. Pravidla je možné si představit jako soubor podmínek. Jestliže jsou všechny podmínky splněny, tak se může vykonat určitá akce. Například, pro daný modul, mohou existovat dvě podmínky. První podmínka se zabývá tím, jestli modul existuje v určité lokaci. Druhá podmínka kontroluje, zdali je místo vedle modulu prázdné. Jestliže jsou podmínky splněné, tak se modul posune určitým směrem. Moduly nemají hlavní jednotku, která řídí celou situaci, ale moduly přemýšlí samy současně. Výhodou tohoto systému je možnost řízení obrovského množství modulů, např. více jak milión. Pokud bychom využívali centralizované řízení, narazili bychom i na omezení možností sběrnice. [1]

### 2.7.3 Komunikace mezi moduly

Komunikace mezi moduly může být dosažena pomocí globální sběrnice – např. CAN sběrnice (Controller Area Network) - populárním automobilovým a později robotickým protokolem. Komunikace lokální mezi jednotlivými moduly může být zrealizována použitím infračerveného záření. Hodně systémů používá obě formy komunikace (Polybot, CKBot, M-TRAN). Bezdrátové technologie je také možno využít podobně jako globální sběrnici. [1]

## 2.8 Příklady robotů

### 2.8.1 M-TRAN III

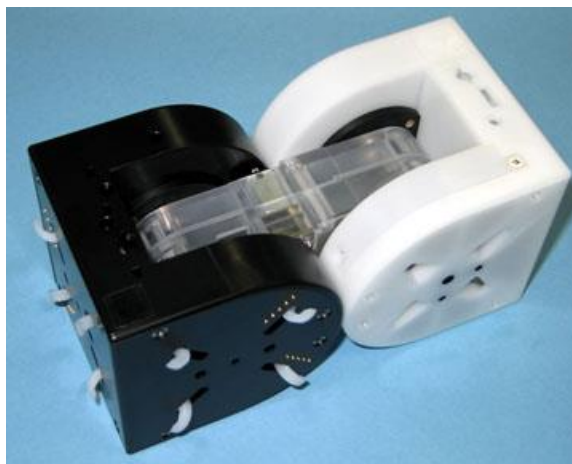
M-TRAN je sebekonfigurovatelný modulární robot, který je vyvíjen AIST a Tokyo-Tech od roku 1998. M-TRAN moduly mohou vytvořit:

- 3D strukturu, která mění vlastní konfiguraci
- 3D strukturu generující menší roboty

M-TRAN systém může měnit svoji strukturu i pohyby podle různého prostředí. V jedné kombinaci může chodit jako krácející robot, poté se přeměnit na tvar hada (Obr. 7), aby mohl proniknout do úzkých prostorů. Při větším množství modulů je možné šplhat na vyšší místa díky přemisťování modulů jednoho za druhým. Možné aplikace M-TRAN robotu jsou autonomní průzkum v neznámém prostředí, nebo záchranné operace v oblastech po katastrofě.[5]



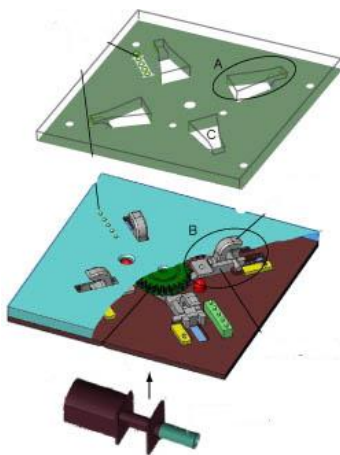
Obr. 7 Přeměna kráčejícího robota na tvar hada [5]



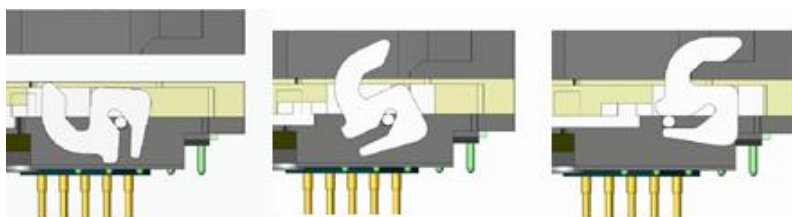
Obr. 8 Vzhled M-TRAN modulu [5]

Konstrukce M-TRAN modulu spojuje výhody modulárních robotů – mřížového typu a řetězového typu. M-TRAN modul se skládá ze dvou hlavních částí, z nichž každá má jednu polovinu krychlovou a druhou válcovou. Tyto dvě části jsou spojeny spojovacím článkem (Obr. 8). Každá ze tří ploch na bloku může být mechanicky propojena s jiným blokem ve všech čtyřech směrech. Propojení je řízeno samotným modulem.

Dřívější verze M-TRAN (M-TRAN II) využívala pro propojení modulů permanentní magnety a aktuátory ze slitiny s tvarovou pamětí, ale kvůli časové a energetické nevýhodnosti vyplynulo, že je rychlejší a energeticky výhodnější použít mechanické zajištění. Připojovací plocha je na obrázku (Obr. 9). Obrázek (Obr. 10) zobrazuje detail zajištění jednotlivých modulů. [5]



Obr. 9 Připojovací plocha modulu [5]



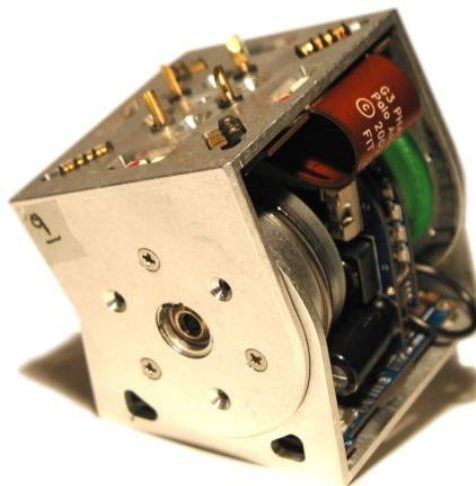
Obr. 10 Detail propojení jednotlivých modulů [5]

Každý modul má čtyři mikrokontroléry, jeden hlavní a ostatní podřazené. Všechny hlavní mikrokontroléry jsou vzájemně propojeny pomocí CAN sběrnice, díky které mohou komunikovat a tak synchronizovat svoje pohyby a řídit se. [5]



### 2.8.2 Polybot G3

Třetí generace modulu (modul G3) má rozměry 50x50x45 mm (Obr. 11). Hlavní motor je vyrobený na zakázku. Využívá modifikovaný bezkartáčový motor Maxon o průměru 32 mm s planetovou převodovkou. Tento nový pohon váží pouze 70 gramů a díky tomu je celková váha modulu přibližně 200 gramů. Motor dodává 1 Nm krouticího momentu a vyrobený hliníkový rám má rozsah pohybu od +90 do -90 stupňů. [9]



Obr. 11 Polybot G3 [10]

Dvě konektorové desky na každé straně modulu jsou identické, hermafroditické a mají 4 možné způsoby rotační symetrie. To znamená, že každé vzájemné propojení může být realizováno s 90 stupňovým inkrementem. Při spojování desek čtyři rýhované kolíky vstupují čtyřmi dírami do protější strany a jsou uchyceny aretačním mechanismem, který může být později uvolněn tvarově paměťovou slitinou. Každá deska má čtyři hermafroditické elektrické konektory pro připojení napájení (tři jsou redundantní) a komunikaci z modulu do modulu. Každá deska má také čtyři infračervené LED diody a senzory pro přesné propojování modulů během rekonfigurace a pro základní modulovou komunikaci. Tato komunikace je použita během inicializace, kdy robot objevuje svoji konfiguraci. [9]

Každý modul obsahuje vestavěný procesor Motorola PowerPC 555 s 1 megabytem externí RAM paměti. Protože je tento výkonný procesor umístěný v každém modulu, tak svoje možnosti ani nevyužije. Celkový cíl autonomního řízení však požaduje využití takového procesoru a paměti. Každý modul komunikuje přes lokální sběrnici využívající CAN (controller area network) standard. Snímání zahrnuje Hallovy senzory zabudované do bezkartáčového stejnosměrného motoru. Natočení je snímáno s rozlišením 0,04 stupňů. Absolutní pozice úhlu senzoru (potenciometru) je také snímána spolu se čtyřmi akcelerometry (jedním redundantním, který může být užit pro úhlové snímání a pro měření orientace relativně ke gravitačnímu zrychlení). Dalšími senzory jsou nárazníky a nízko rozlišovací silové senzory na spojovacích kolících, kvůli potenciálnímu kontaktu. Díky umístění infračervených komponentů na stěny modulu G3, mohou být také tyto komponenty použity pro snímání vzdálenosti. [9]

### 2.8.3 ATRON

Základním návrhem se stala mřížová struktura robota téměř kulových modulů (Obr. 12). Každý modul je tvořen dvěma částmi, které se proti sobě otáčí. Každá hemisféra má dva

konektory s vidlicí a dva konektory se zásuvkou pro přichycení dalšího modulu. Každý konektor má infračervený senzor vzdálenosti a také senzor natočení. Rotační kloub je namáhán silovým a momentovým zatížením ze všech směrů. Proto je zde využito ložisko, které tyto síly zachytí. [11]

Transformace je řešena přesouváním modulů rotacemi o násobku  $90^\circ$ , přichycením k dalšímu modulu, a uvolněním přechozího držení. Často je potřeba několik sekvencí, aby se robot dostal do cílové pozice. Aby nebyl problém s překroucenými kabely uprostřed modulu, byl navržen kroužek, který umožňuje přenášet signál s nekonečnou rotací. Z praktických důvodů bylo lepší, aby přenášených signálů bylo co nejméně, proto má každá hemisféra svůj mikrokontrolér. Mezi moduly prochází pouze napájení a sériová komunikace. Přenáší se tedy pět kabelů – 10 V, 5 V, uzemnění a dva RS485 signály. Kroužek je dále také využíván jako senzor absolutního natočení s optickým enkodérem. Pokud tedy dojde k natočení modulu, i přestože ho modul sám nevyvolá (např. vnějším silovým působením), může na to mikrokontrolér zareagovat a vyhnout se chybám. Obrázek (Obr. 13) ukazuje kroužek pro přenos dat. Také je možné vidět planetovou převodovku. Na okraji modulu je ložisko, které zachycuje zatěžující síly. [11]



Obr. 12 Atron robot [11]



Obr. 13 Atron vnitřek modulu [11]



## 3 KONSTRUKČNÍ NÁVRH

### 3.1 Požadavky na konstrukční návrh

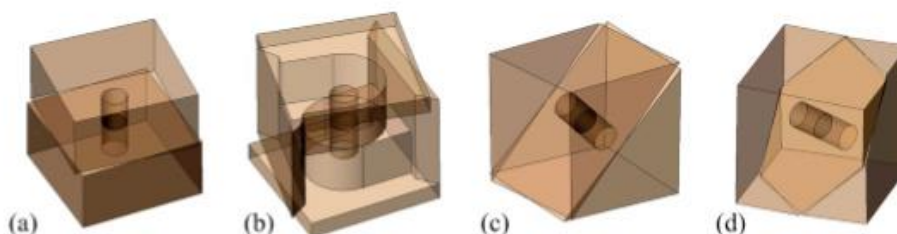
Je nutné zvolit požadavky, které má konstrukční návrh splňovat. Na základě těchto požadavků potom může vzniknout vhodný konstrukční návrh. Byly definovány tyto požadavky:

- Vytvoření různých struktur (aby moduly byly schopné vytvořit strukturu, kdy jsou poskládány sériově za sebou jako had, potom aby se dokázal propojit poslední modul s prvním modulem (lokomoce), a také aby bylo možné připojit modul do boku hada vytvořit tak strukturu připomínající pavouka)
- Jednoduchá struktura (nejlépe aby natáčení modulů bylo realizováno na sebe kolmými rotačními osami pro jednodušší představitelnost pohybů)
- Ekonomicky dostupné, jednoduše vyrobitelné (kvůli realizovatelnosti), pro vnitřní použití

### 3.2 Volba struktury

Byla zvolena řetězová struktura. Řetězová struktura má tu výhodu, že i při menším počtu modulů je možné dosáhnout mnoha různých prostorových pohybů a možností rekonfigurace.

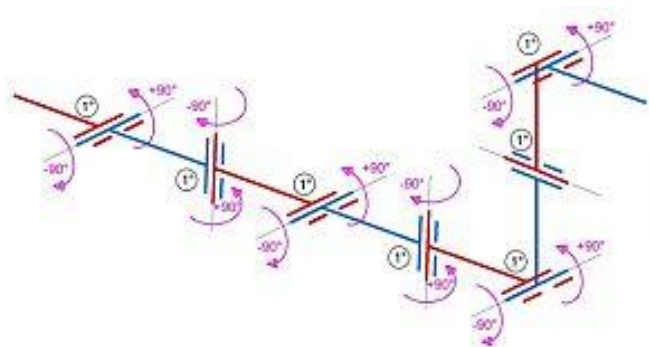
Dále bylo nutné zvolit, jak budou rotační osy v modulu situovány. Pro symetrické tvary modulů existuje několik možných řešení (Obr. 14). Z těchto řešení vyplývá, že nejjednodušší na představitelnost při použití více modulů je řešení, které má osu kolmou ke stěnám modulu (řešení *a* a řešení *b*). Proto byla vybrána takto situovaná osa rotace.



Obr. 14 Symetrické tvary modulů [12]

Z ekonomického hlediska je výhodnější spojit dva symetrické moduly pevně k sobě a to tak, aby byly osy rotace navzájem kolmé. Díky tomuto spojení stačí pouze jeden mikrokontrolér a jedno napájení na dvě rotace. Odpadá nutná realizace automatického spojování a rozpojování mezi těmito dvěma částmi. Pro pohyby robota jsou dvě na sebe navzájem kolmé rotace velmi dobře využitelné. To potvrzuje podobné kinematické schéma hada, které je vidět na obrázku (

Obr. 15). V reálném světě potom had vypadá jako na obrázku (Obr. 16). [13]



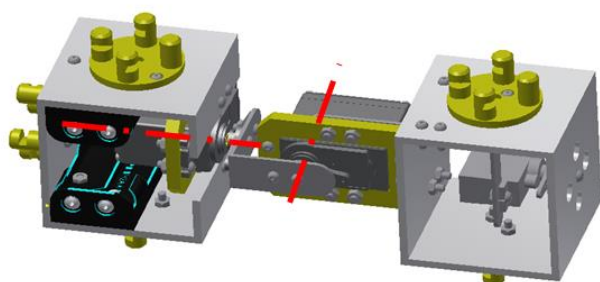
Obr. 15 Kinematické uspořádání robota [13]



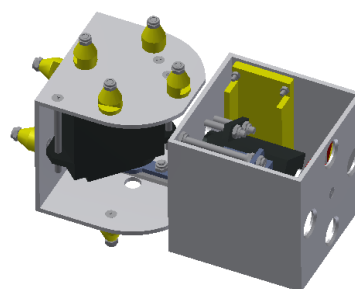
Obr. 16 Fotografie robota hada [13]

Díky vzájemným kolmým osám rotace je poměrně jednoduše představitelné, jaký pohyb nastane při natočení jednotlivých os. Díky tomu je programování pohybů robota poněkud snazší a některých triviálních poloh lze dosáhnout i bez složitých výpočtů.

Výsledkem této kinematické varianty jsou dva konstrukční návrhy (Obr. 17), (Obr. 18).



Obr. 17 Konstrukční návrh (varianta a)



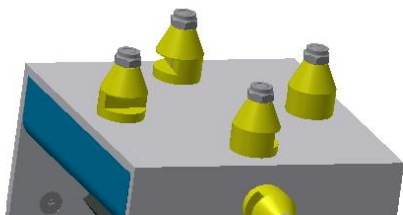
Obr. 18 Konstrukční návrh (varianta b)

Volba výsledného konstrukčního návrhu:

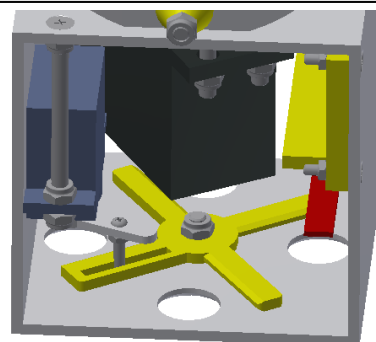
Podle schopnosti rekonfigurace je výhodnější varianta *b*. Toto uspořádání umožňuje měnit strukturu i s menším počtem modulů. Modul je také o 1/3 kratší a je tak schopný rekonfigurace na menším prostoru. Nevýhodou je ovšem omezenější prostory pro potřebné vybavení Molecube, nicméně bylo ověřeno, že je možné do prostoru umístit všechny potřebné součásti.

### 3.3 Spojení dvou modulů

Automatické spojení dvou modulů probíhá tak, že nejprve se připojovací kužely (Obr. 19) jednoho modulu zasunou do děr druhého modulu. Poté se zajišťovací kříž natočí a zasune se do drážek v kuželech a zajistí tak modul proti vysunutí (Obr. 20). Zajištění tímto způsobem je výhodné kvůli tomu, že během spojení není již potřeba žádná další elektrická energie. Kříž je uchycený samostatně šroubem ke spodní části tak, aby se mohl natáčet.



Obr. 19 Připojovací kužely



Obr. 20 Připojovací kříž

Bylo nutné zvolit, jakým způsobem zajistit pohon pro natáčení zajišťovacího kříže. Pro tuto aplikaci by bylo vhodné využít buď mikro servomotor Tower Pro, nebo elektromagnet.



Obr. 21 Elektromagnet [26]



Obr. 22 Servomotor Tower Pro SG90 [19]

#### Vlastnosti servomotoru [19]

Váha	9 g
Rozměry	22,2*11,8*31 mm
Krouticí moment	1 kg*cm (4,8 V)
Provozní rychlost	0.1 sec/60° (4,8 V)
Provozní napětí	3 až 6 V
Teplotní rozsah	0 °C až 55 °C

#### Vlastnosti elektromagnetu: [26]

Váha	12 g
Rozměry	11*12*20,3 mm
Odpor	32,8 ohm
Provozní napětí a odběr proudu	3 V 80 mA; 6 V 170 mA; 9 V 260 mA; 12 V 350 mA

Shrnutí:

Elektromagnet	Mikro servomotor
<ul style="list-style-type: none"><li>• Jednodušší ovládání</li><li>• Menší rozsah pohybu</li><li>• Obtížnější montáž (nemá připojovací otvory)</li></ul>	<ul style="list-style-type: none"><li>• Větší rozměry</li><li>• Jednodušší montáž</li></ul>

Jako řešení byl zvolen servomotor, hlavně kvůli jednodušší montáži. Řízení je sice složitější (je potřeba mikrokontrolér) ale ten už modul obsahuje, takže složitější řízení zde není problémem.

Na obrázku je vidět detail připojovacího kuželu (Obr. 23). Připojovací kužely jsou na jedné ploše čtyři, aby spolehlivě zachytily silové působení mezi moduly. Kvůli ekonomické úspoře jsou na 3D tiskárně vytisknuty jenom tyto kužely a připojení kuželu ke straně modulu je realizováno šroubem, který prochází středem kuželu. Díky tomuto způsobu je kužel pevnější a odolává lépe silovému působení.



*Obr. 23 Připojovací kužel*

Vzhledem k potřebě využití vnitřního prostoru v horní části modulu na baterie a na servomotor byla zvolena varianta, kdy šrouby jsou orientovány zevnitř ven, aby se celá jejich hlava ukryla v zahlbubení ve stěně modulu. Takto se zabrání kolizi matic a držáku pro baterie a kolizi matice a horního servomotoru. Nevýhodou je, že nad připojovacím kuželem je nyní matice, která je asi o 1 mm delší než pouze hlava šroubu. Bylo však ověřeno, že během automatického spojování k žádné kolizi nedochází, proto bylo využito toto řešení.

### 3.4 Propojení modulů

Důvodem pro propojení modulů může být napájení modulů a komunikace mezi moduly. Propojení modulů je možné realizovat čtyřmi variantami. Způsoby propojení modulů:

1. Propojení kabely s konektory na stěnách modulů sloužící ke sdílení elektrické energie a pro komunikaci mezi moduly
2. Kabely je propojeno napájení a komunikace je bezdrátová
3. Komunikace je vedena kabely a každý modul má vlastní napájení
4. Každý modul má vlastní napájení i bezdrátovou komunikaci

Výběr varianty:

Výběr varianty je na základě tabulky, kde jsou zohledněny jednotlivé vlastnosti variant v závislosti na důležitosti. Váhu 1 až 10 jsem přiřadil jednotlivým aspektům variant, což jsou

pracnost výroby, cena, větší pohyblivost robota (ve smyslu neomezení kabely), spolehlivost robota díky napájení (napájení ze zdroje se mi jeví jako spolehlivější).

Body 1 až 5 byly známkovány jednotlivé varianty – 5 bodů pro nejvíce vyhovující, 1 bod nejvíce nevyhovující.

	Váha	Varianta 1	Varianta 2	Varianta 3	Varianta 4
<b>pracnost</b>	8	1	3	3	5
<b>cena</b>	10	4	3	1	2
<b>pohyblivost</b>	6	2	2	2	5
<b>spolehlivost</b>	5	4	5	1	2
<b>CELKEM</b>		80	91	51	<b>100</b>

Na základě těchto výsledků byla vybrána čtvrtá varianta, totiž že každý modul má vlastní napájení pomocí baterie a bude obsahovat mikrokontrolér pro bezdrátovou komunikaci. Nevýhodou této varianty je sice vyšší cena kvůli bateriím pro každý modul, také je nevýhodou že baterie se může během práce robota vybit a modul přestane fungovat, nicméně nespornou výhodou, na kterou byl kladen důraz, je jednodušší výroba kvůli tomu, že není potřeba umisťovat konektory na každé propojovací stěně modulu. Také je mnohem větší nezávislost robota při pohybu díky bateriím.

### 3.5 Servomotor

Pohyby robota uskutečňují dva servomotory. Bylo vybráno oblíbené modelářské Tower Pro MG946R (Obr. 24) kvůli vysokému poskytovanému krouticímu momentu. Tento servomotor je také vybaven kovovými převody. Cena servomotoru se pohybuje kolem 7 dolarů.



Obr. 24 Servo Tower Pro [18]

Vlastnosti servomotoru: [18]

Váha	55 g
Rozměry	40,7*19,7*42,9 mm
Krouticí moment	13 kg*cm (1,28 Nm)
Rychlost pohybu	0,2 sec/60° (4,8 V)
Provozní napětí	4,8-7,2 V
Teplotní rozsah	0 - 55°C

Stanovení počtu modulů, které tento motor uzvedne, závisí na hmotnosti a rozměrech modulu. Modul bude mít délku 120 mm, a celková hmotnost se skládá ze dvou hlavních servomotorů (2\*55 g), jednoho zajišťovacího servomotoru (9g), baterií (30g) a z hliníkového rámu. Naměřená přesná váha celého modulu je 364 g.

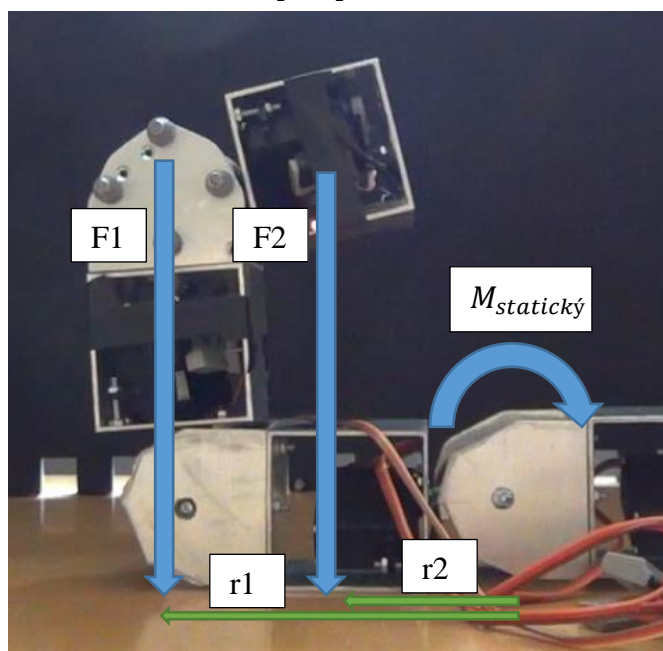
Pro zjednodušení výpočtu je možné uvažovat těžiště přibližně uprostřed modulu. Gravitační síla působící na jeden modul je spočítána podle vztahu (3.1) a má velikost 3,53 N.

$$F_g = m \cdot g = 0,36 \cdot 9,81 = 3,53 \text{ [N]} \quad (3.1)$$

Aby moment motoru byl dostačující na uzvednutí ostatních modulů, musí být větší než statický a dynamický moment (3.2). Pro rekonfiguraci ve vertikální rovině, kdy se koncový modul propojuje se čtvrtým modulem, byla nalezena nejnepríznivější varianta a pro tu byl proveden výpočet, jestli je možné tuto rekonfiguraci provést. Statický moment pro tuto variantu je vypočítán podle vztahu (3.3). Obrázek k výpočtu ukazuje umístění sil a vzdáleností od osy rotace (Obr. 25).

$$\begin{aligned} M_{motoru} &\geq M_{statický} + M_{dynamický} \text{ [Nm]} \\ M_{motoru} &\geq M_{statický} + I \cdot \varepsilon \text{ [Nm]} \end{aligned} \quad (3.2)$$

$$M_{statický} = F_1 \cdot r_1 + F_2 \cdot r_2 = 5,3 \cdot 0,12 + 3,53 \cdot 0,06 = 0,85 \text{ [Nm]} \quad (3.3)$$



Obr. 25 Silové působení na moduly

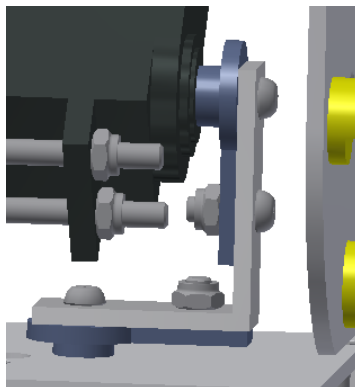
Počáteční zrychlení, kterým se moduly začnou pohybovat, je vypočítáno podle vztahu (3.4). Moment setrvačnosti je vztažen k ose rotace (ose natáčejícího se servomotoru). Toto zrychlení je pro vyvození pohybu naprosto dostatečné. V reálném experimentu bude velikost počátečního zrychlení o něco menší, neboť ve výpočtu jsou zanedbány ztráty (např. třením v převodech). Je nutné také si uvědomit, že během pohybu se bude statický moment zmenšovat a pokud se tedy soustava začne pohybovat, už je jisté, že servomotor je schopen provést tuto rekonfiguraci. Z výpočtu (3.4) tedy vyplývá, že soustava se začne pohybovat



s poměrně velkým úhlovým zrychlením a rekonfigurací je tedy možné provést. Později byla tato rekonfigurace ověřena experimentem (7.14.3).

$$\varepsilon = \frac{M_{motoru} - M_{statický}}{I} = \frac{1,28 - 0,85}{0,03377} = 12,73 \text{ [rad/s}^2\text{]} \quad (3.4)$$

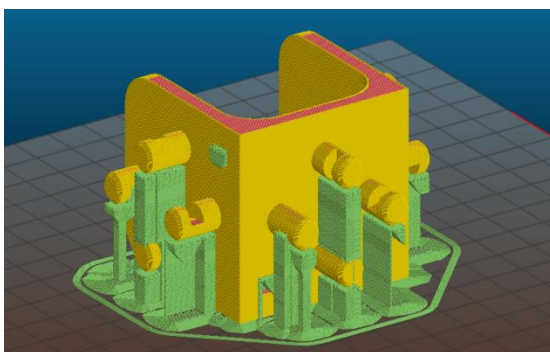
K servomotoru jsou dodávány jako příslušenství páky pro snadné připojení servomotoru k požadované aplikaci. Tyto páky bylo nutné zkrátit, aby zabíraly co nejméně místa při otáčivém pohybu (Obr. 26). Díky této úpravě je možný průchod kabelů v těsné blízkosti středu modulu a také umístění baterií v horní části modulu.



Obr. 26 Zkrácené páky servomotoru

### 3.6 Výroba modulu

Jednou z možností výroby je vytisknutí celé nosné části modulu na 3D tiskárně. Toto řešení je ideální z hlediska pracnosti během montáže, ale cenově je poměrně drahé. Jedna strana modulu by stála asi 180 Kč. Jak by výrobek po tisku vypadal, je vidět na obrázku (Obr. 27). Části, které mají zelenou barvu, by se po tisku odlomily, ale pro tisk jsou důležité, neboť tiskárna nedokáže tisknout do prázdného prostoru.



Obr. 27 Modul po 3D vytisknutí



Obr. 28 Hliníkový profil [23]

Kvůli snížení ceny byla zvolena možnost využití hliníkového profilu. Vybraný hliníkový profil má rozměry 60x60x2 mm (Obr. 28). Hlavní výhodou je, že profil tvoří nosnou konstrukci modulu a na 3D tiskárně se vytisknou jen připojovací kužely.

### 3.7 Kontrola spojení modulů

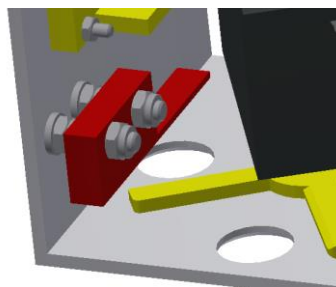
Aby mohla rekonfigurace probíhat autonomně, je důležité mít jistotu, že jsou moduly propojené. Tato kontrola se dá zajistit senzory. Pro modul byly navrženy tři možnosti, jaké senzory použít:

1. Využít Hallovu sondu, která by byla přichycena na spodní část modulu. Horní modul by pak měl na každé straně, kde jsou připojovací kužely, uprostřed uchycený neodymový magnet, takže pokud by se moduly spojily, Hallova sonda by zjistila přítomnost magnetu. Velkou nevýhodou této varianty je nutná instalace tří magnetů i Hallovy sondy.
2. Využití skutečnosti, že moduly jsou připojeny ve chvíli, kdy připojovací kužely jsou uvnitř spodního modulu. Je tedy možnost zjišťovat pouze přítomnost těchto kuželů. Tím se nám problém sníží pouze na jeden snímač. Realizace je možná pomocí infračerveného vysílače a přijímače. Před kuzelem by byla dioda (vysílač) emitující záření v určitém kmitočtu (aby došlo k odfiltrování denního světla) a za kuzelem by byl přijímač. Pokud by kužel nebyl v připojovacím místě, tak paprsek již nemůže k přijímači a dojde tak k přerušení paprsku. Nevýhodou této varianty je poměrně složité spolehlivé uchycení vysílače a přijímače a také je nebezpečnost vzájemného rušení mezi moduly.
3. Nejjednodušší varianta je využít mikrospínač (Obr. 29), jehož hmatník se mechanicky ohne po vložení připojovacího kuželu. Tato varianta využívá mechanické části, takže je podle výrobce možných asi 10.000 cyklů, nicméně pro modul je to dostatečně vysoká hodnota. [30]

Na základě toho, že mikrospínač je spolehlivý a velmi jednoduchý na montáž, že stačí pouze dva připojovací kabely a že 10 000 cyklů je pro robota dostačující, byla vybrána varianta 3. Mikrospínač je umístěn na stěně modulu a je přichycen dvěma šrouby M3. Hmatník pak je umístěn přesně nad středem otvoru, kterým je do modulu vsouván připojovací kužel dalšího modulu (Obr. 30).



Obr. 29 Mikrospínač [30]



Obr. 30 Uchycení kontrolního mikrospínače

### 3.8 Uchycení vývojové desky

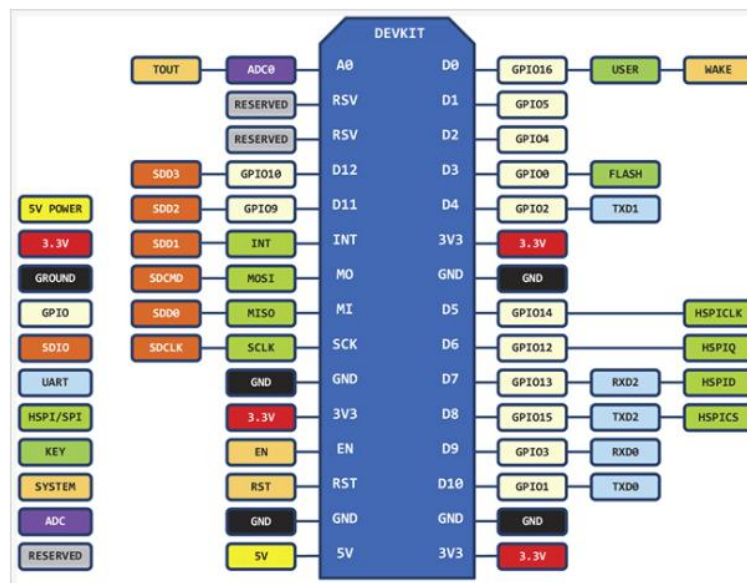
Vývojová deska, která obsahuje řídicí mikrokontrolér, má pro uchycení čtyři otvory o průměru 2,5 mm. Šroub M3 by otvorem prošel jen obtížně, proto byla deska uchycena ke stěně čtyřmi šrouby M2. Vývojová deska není opatřena žádným obalem nebo krabičkou, která by zabránila případnému zkratu po dotyku se stěnou modulu, proto bylo nutné mezi stěnu a desku přilepit vrstvu izolační pásky. Uchycení ke stěně je realizováno tak, aby byl přístupný micro-USB port, kterým je možné nahrávat nový program do modulu nebo modul napájet. Dále je nutné mít přístup k alespoň ke čtyřem digitálním vstupům/výstupům pro řízení servomotorů a pro připojení výstupu z mikrospínače (Označené GPIO (Rozložení vstupů a



výstupů je na Obr. 31)), dále k uzemnění mikrospínače (GND) a k externímu napájení +5V a uzemnění (GND), pro napájení z baterie. Mikrokontrolér je tedy umístěný vhodně tak, že splňuje jak přístup k USB konektoru, tak k potřebným výstupům a vstupům.

Do vývojové desky NodeMCU je tedy připojen:

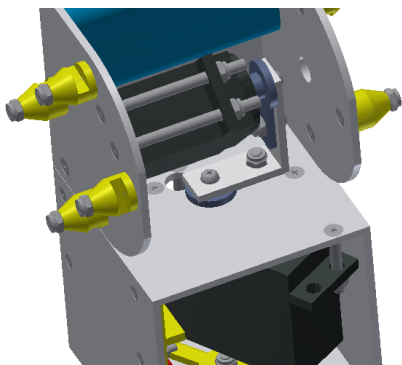
- GPIO16 – horní servomotor
- GPIO5 – dolní servomotor
- GPIO4 – zajišťovací servomotor
- GPIO0 – mikrospínač



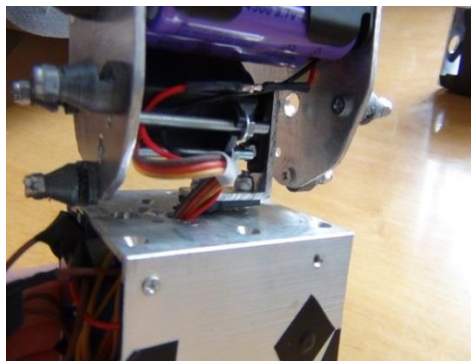
Obr. 31 Umístění vstupů a výstupů na Vývojové desce [31]

### 3.9 Průchod kabelu mezi horní a dolní částí modulu

Modul se skládá z horní a dolní části. V horní části je servomotor a baterie, v dolní části dva servomotory a vývojová deska. Je potřebné kabely propojit horní a dolní část. Velkou výhodou je, že servomotory mají jen omezený rozsah pohybu +90 až -90 stupňů. Díky tomuto omezenému rozsahu není pro přenos nutno uvažovat sběrné kartáče nebo nějaké podobné řešení, které řeší neomezenou rotaci. Nicméně pohyb horní části Molecube je ve dvou osách – jednak se naklápí a jednak i natáčí. Je patrné, že čím blíže středu otvor pro průchod kabelu bude, tím menší bude i rozdíl potřebné délky kabelu pro všechny varianty pohybu horní části modulu. Nejvhodnější místo bylo zvoleno co nejbližší středu v jedné čtvrtině spodního modulu. V tomto místě nekoliduje ani natáčecí páka servomotoru, ani do něj nezasahuje horní servomotor při maximální naklopení (Obr. 32). Délka kabelu byla nastavena na nejdelší variantu a kabely servomotoru jsou spojeny dohromady s kabely od baterií. Při této konfiguraci robot během natáčení o kabel nekoliduje, ani se nestává, že by kabel bránil v plném rozsahu pohybu robota (Obr. 33).



*Obr. 32 Modul v krajní poloze a maximálně natočen*



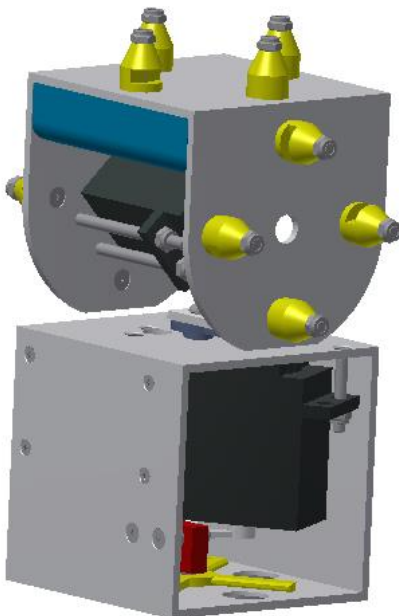
*Obr. 33 Umístění otvoru pro průchod kabelu*

### 3.10 Použité matice

Modul vykonává různé pohyby, proto obvyklé matice bez zajištění nejsou pro tuto aplikaci vhodné, neboť se kvůli otřesům povolují a poté může modul ztratit svoji funkčnost. Byly zvoleny matice samojistící, které potřebují k povolení určitý moment, který během otřesů nenastane, takže touto volbou je zajištěné stálé utažení šroubových spojů.

### 3.11 Uložení baterií

Vhodné místo pro baterie se nachází v horním modulu. Celý původní držák baterií se do prostoru nevejde, proto ho bylo nutné uprostřed o 3mm zkrátit. Baterie jdou do držáku stále vložit a držák se již vejde mezi stěny modulu. Držák pojme dvě baterie. Na celkovém pohledu je vidět uspořádání komponentů v modulu, držák baterií má světle modrou barvu (Obr. 34).



*Obr. 34 Celkový pohled na modul*

## 4 VOLBA MIKROKONTROLÉRU A NAPÁJENÍ

### 4.1 Volba mikrokontroléru

Pro řízení servomotorů je potřebné zvolit vhodný mikrokontrolér. Tento mikrokontrolér bude mít také za úkol komunikovat s centrální jednotkou, aby byly pohyby robotu synchronizované. Pro tuto aplikaci je vhodné využít buď mikrokontrolér Arduino, který má velkou zákaznickou podporu, nebo Wi-fi modul se schopnostmi mikrokontroléru NodeMCU.

#### 4.1.1 Mikrokontrolér Arduino

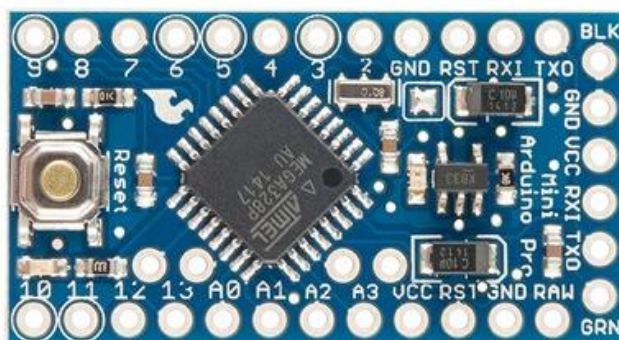
Arduino je open-source platforma, která se dá jednoduše používat v oblasti hardware a software. Arduino desky jsou schopné číst vstupy např. od světelných senzorů, tlačítek nebo Twittrových zpráv a poté je převést na výstupy. To mohou být motory, LED světla nebo online zprávy. Arduino vykonává program podle instrukcí nahraných na mikrokontroléru. Ty se mohou nahrát na desku díky programovacímu jazyku (založeným na Wiring) a díky Arduino softwaru (IDE), založeném na Processingu. [16]

Několik let se již Arduino používá jako řídicí jednotka v tisíci projektech, od běžných zařízení po komplexní vědecké nástroje. Celosvětová komunita studentů, vědců, nadšenců, umělců, programátorů a profesionálů shromáždila již obrovské množství dostupných vědomostí, které při aplikacích značně pomáhají. Arduino pochází z Ivera Interaction Design Institute, kde byl vytvořen jako pomůcka pro studenty pro rychlé prototypování, pro studenty s malými znalostmi v oblasti elektroniky a programování. [16]

Výhody mikrokontroléru: [16]

- Funguje na Windows, Linux
- Jednoduché programovací prostředí
- Open source a rozšiřitelný software – Je možné využít C++ knihovny
- Open source a rozšiřitelný hardware – Je možné vytvořit vlastní verzi modulu

Mikrokontrolér se vyrábí v mnoha provedeních. Pro Molecube je nejvhodnější verze Pro Mini (Obr. 35), která je dostatečně výkonná a má kompaktní rozměry. Nelze připojit přímo k počítači, ale program se musí nahrát pomocí dalšího zařízení. Díky tomu se ale šetří prostor na mikrokontroléru.

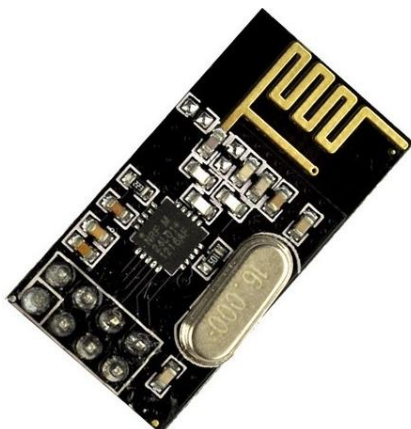


Obr. 35 Mikrokontrolér Arduino Pro Mini [17]

Vlastnosti Mikrokontroléru: [17]

Mikroprocesor:	ATmega328
Provozní napětí:	5V
Vstupní napětí:	5-12 V
Digitální I / O piny:	14 (z toho 6 poskytuje PWM výstup)
Analogový vstup:	6 pinů
DC proud na každý I / O Pin:	40 mA
Flash paměť:	32 KB (z toho 2 KB používá zavaděč)
SRAM:	2 KB
EEPROM:	1 KB
Takt procesoru:	16 MHz
Délka:	30 mm
Šířka:	18 mm

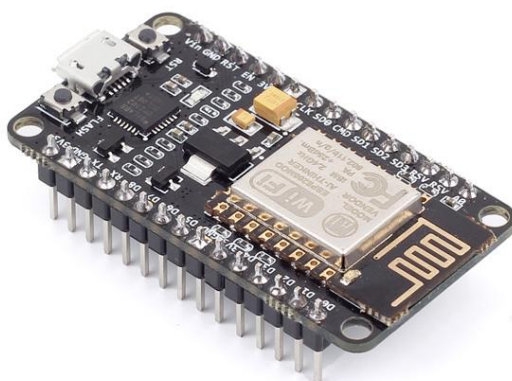
Vlastní mikrokontrolér nemá možnost bezdrátové komunikace. Proto je nutné připojit např. modul pro bezdrátovou komunikaci na rádiových vlnách (Obr. 36). Modul má implementovanou anténu s dosahem na volném prostranství až 100 m, což pro Molecube dostačující.



Obr. 36 Bezdrátový modul NRF24L01 [21]

#### 4.1.2 Vývojová deska NodeMCU

Vývojová deska obsahuje Wi-fi čip ESP8266 se schopnostmi mikrokontroléru. Tato vývojová deska má již vyvedené výstupní a vstupní kolíky sloužící pro připojení externích zařízení (v našem případě připojení servomotorů) a dalších senzorů (Obr. 37).



Obr. 37 Vývojová deska NodeMCU [29]

Wi-fi modul je možné programovat pomocí prostředí pro programování Arduina. V článku NodeMCU Wi-fi modul s ESP8266 na stránkách Arduinotech se popisuje, jak je možné implementovat programování tohoto modulu do vývojového prostředí pro Arduino. [28]

Vlastnosti vývojové desky NodeMCU: [27]

Procesor	ESP8266
Napájení	USB
Paměť	20 kBytů
Wi-fi standard	802.11 b/g/n
Možnosti Wi-fi připojení	Wi-Fi Direct (P2P) – přímá vzájemná komunikace desek soft-AP – vytvoření přístupového bodu

Na základě těchto informací o mikrokontroléru Arduino Pro Mini a o vývojové desce NodeMCU je patrné výhodnější využít desku NodeMCU, zvláště kvůli již implementované bezdrátové komunikaci. Wi-fi připojení je také vhodnější, protože může komunikovat přímo s notebookem nebo mobilním telefonem vybaveným Wi-fi zařízením. Pohodlné je také USB připojení pro nahrávání programů. Díky implementaci do vývojového prostředí pro Arduino, kde je možné programovat v jazyce C, odpadá problém s programováním modulu.

## 4.2 Napájení robota

### 4.2.1 Požadované napětí a proud

Vývojová deska NodeMCU obsahuje modul ESP8266, který pracuje při napětí 3,3 V. Deska již obsahuje vlastní měnič napětí, takže je modul možné napájet přímo 5 V. Proud potřebný pro mikrokontrolér se mění podle vykonávané činnosti. Orientačně lze ale říci, že pokud je deska napájena z USB, tak pro USB je maximální proud 500 mA. To znamená, že by mikrokontrolér neměl mít určitě vyšší spotřebu proudu.

Dále je nutno napájet 2 velké servomotory Tower Pro a jeden mikro servomotor. Ideální napětí pro servomotory je 5 - 6 V. Podle zkušeností uživatelů je vhodné mít k dispozici zdroj schopný poskytovat přibližně jeden ampér na jeden velký servomotor.

Molecube modul se skládá ze dvou velkých servomotorů, jednoho malého a jednoho mikrokontroléru. Odběr, který by měl být teoreticky maximální pro Molecube je přibližně 4 A. Napětí mezi 5 až 6 V.

#### 4.2.2 Návrh baterií

Pro nová zařízení se v současné době jeví jako nejvýhodnější využít technologie Li-ion nebo Li-po.

Lithium-iontové Baterie:

Lithium-iontové baterie se vyvíjí od roku 1912. Populárními se staly až v roce 1991, kdy je vyvinula společnost Sony. Lithium-iontové baterie mají vysokou energetickou hustotu a stojí méně než Lithium-polymerové baterie. Výhodou je velmi pomalé samovybíjení. Nicméně tyto baterie postihuje stárnutí – ať už jsou využívány nebo nejsou (Obr. 38). [25]

Lithium-Polymerové Baterie:

Lithium-polymerové baterie jsou datovány od roku 1970. Jejich první design zahrnoval suchý pevný polymerový elektrolyt, který se podobá plastickému filmu. Proto tento typ baterií má schopnost mít tvar kreditní karty se stálou schopností výdrže baterie. Kromě toho lithium-polymerové baterie jsou velmi lehké a mají vylepšenou bezpečnost. Nicméně tyto baterie stojí výrobně více a mají menší energetickou hustotu než lithium-iontové baterie (Obr. 39). [25]



Obr. 38 Li-ion baterie 14500 [32]



Obr. 39 Li-po baterie [33]

Srovnání baterií: [32][33]

Baterie	Li-ion 14500	Li-po
Nominální napětí	3,7 V	3,7 V
Maximální proudový odběr	2C (4,6 A)	25C (21 A)
Rozměry	13.5–14.5 x 50.5 mm	25*8.5*40 mm
Kapacita	2300 mAh	850 mAh
Přibližná cena (ebay.com)	40 Kč	60 Kč

Rozměry obou baterií (objemově) jsou srovnatelné. Li-ion se vyznačuje přibližně třikrát větší kapacitou baterie. Li-po zase několikanásobně větší schopností poskytovaného



proudu. Vzhledem k tomu, že Li-ion baterie 14500 má stejnou velikost jako AA baterie a proud potřebný pro modul Molecube (maximálně 4 A) je postačující, je výhodnější využít baterie Li-ion.

Modul bude osazen dvěma bateriemi Li-ion 14500 řazenými sériově (nominální napětí 7,4 V). Poté bude nutné využít stabilizátor napětí, kterým bude napětí sníženo na 5V a budou jím napájeny jak motory, tak i řídicí vývojová deska.

#### 4.2.3 Držák baterie

Modul je napájen dvěma bateriemi typu 14500, velikostí podobné AA bateriím. Pro uchycení baterií je využit držák baterií pro dvě baterie (Obr. 40).

#### 4.2.4 Změna velikosti napětí

Protože servomotory určují svoje natočení podle potenciometru, je potřebné motory napájet stále stejným napětím. Tím je zaručené, že natočení a všechny pohyby proběhnou vždy v pořádku. Kvůli napájení z baterií, kdy se hodnota mění vybíjením baterie, je potřebné využít stabilizátor stejnosměrného napětí. Byl zvolen spínací stabilizátor kvůli své vysoké účinnosti (Obr. 41).



Obr. 40 Držák baterie AA [20]



Obr. 41 Stabilizátor napětí DSN6009 [22]

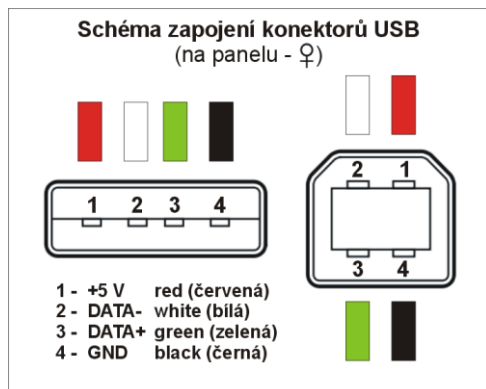
Vlastnosti stabilizátoru: [22]

Vstupní napětí	3V ~ 32V
Výstupní napětí	5V ~ 35V
Proud	4A
účinnost	Až 94%
Rozměry	43*21*14 mm

#### 4.2.5 Napájení modulu z externího zdroje

Během testování bylo nutné robot testovat i při napájení z externího zdroje. Tím může být stejnosměrný zdroj napětí nebo externí baterie, jako Power banka. První pokusy o napájení z Power banky nebyly úspěšné, servomotory se stále trásly. Také po rotačním pohybu do požadované polohy docházelo ke kmitání kolem požadovaného úhlu. Často se také stávalo, že došlo i samo-resetování mikrokontroléru.

Nejprve bylo vyzkoušeno připojit kondenzátor, který by mohl vylepšit chování Molecube, ale zlepšení bylo pouze nepatrné. Vzhledem k tomu, že Power banka i servomotory byly v pořádku, bylo zjištěno že problém musí být ve spojovacích vodičích. První testy probíhaly pro jednoduché prototypování na nepájivém kontaktním poli, to ale není pro vyšší proudy (4 A) vůbec vhodné. Tak bylo vyřazeno z obvodu. Poté bylo ověřeno, že USB kabel propojující Power banku a Molecube není vhodný pro přenos vyšších proudů než 500 mA. Proto byl vyroben vlastní kabel, který se skládá z USB konektoru a dvou vodičů o průřezu 1 mm<sup>2</sup>. Připojení k USB konektoru bylo na +5 V a na GND podle obrázku (Obr. 42). Poté po těchto úpravách již modul Molecube funguje bez problémů i při externím napájení.



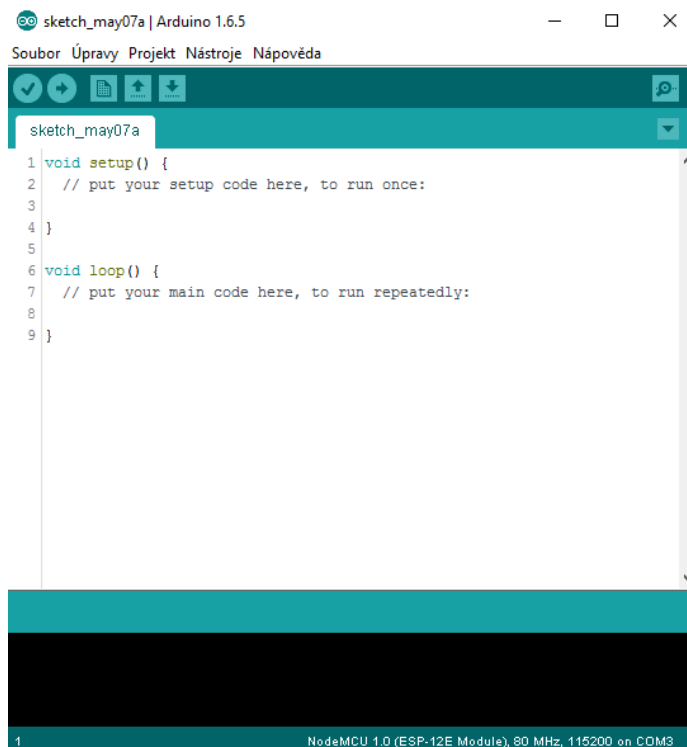
Obr. 42 USB schéma [34]



## 5 PROGRAMOVÁNÍ MIKROKONTROLÉRU

### 5.1 Vývojové prostředí

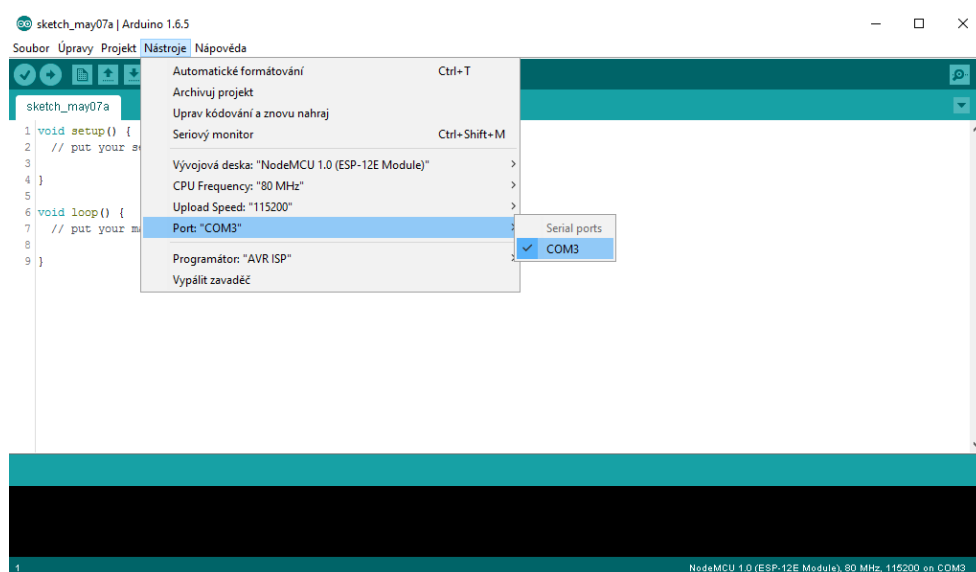
Mikrokontrolér lze programovat ve vývojovém prostředí pro Arduino v jazyku C (Obr. 43). Vývojové prostředí lze stáhnout ze stránek [arduino.cc](http://arduino.cc). Na stránkách je k dispozici hned v několika verzích, tento projekt je realizován ve verzi 1.6.5. Program je tvořen dvěma hlavními smyčkami – první smyčka je nastavovací a ta se vykoná pouze jednou. V této smyčce se definuje např. na jakých vstupech jsou připojeny servomotory. Druhá smyčka se provádí stále dokola. Z této smyčky se také neustále můžeme odkazovat do vytvořených podprogramů.



Obr. 43 Vývojové prostředí

### 5.2 Připojení NodeMCU

Deska NodeMCU se po připojení micro-USB kabelem k počítači stane aktivní a připojení je možné zkontrolovat po kliknutí na *Nástroje*, *Port* a deska je dohledatelná pod některým portem (Obr. 44). Číslo portu závisí na použitém USB portu v počítači. Nastavení pro nahrání programu by mělo být stejné jako je na obrázku: vybraná vývojová deska je *NodeMCU 1.0*, *CPU 80 MHz*, *Upload Speed 115200*.



Obr. 44 Výběr sériového portu

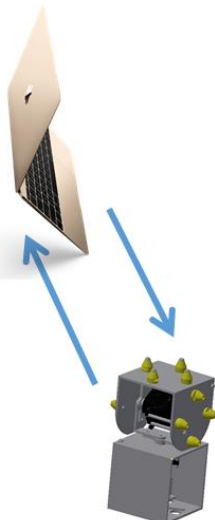
### 5.3 Terminologie používaná v sítích

Router (směrovač) zajišťuje propojení všech zařízení do jedné sítě. Řídí vzájemnou komunikaci mezi zařízeními a obvykle také umožňuje zařízením v síti přístup k internetu. Zařízení můžeme připojit k routeru buď pouze kabely (hovoříme o lokální počítačové síti LAN), nebo je možné využít bezdrátové připojení (WLAN – Wireless Local Area Network). Obvykle se mezi uživateli nazývá tato síť Wi-fi síť. Pro možnost bezdrátového připojení dalších zařízení musí router fungovat jako Access Point (přístupový bod). Poté může i bez dalších zařízení vytvořit vlastní Wi-fi síť. Dříve byl router snadno rozpoznatelný kvůli jedné nebo více anténám, dnes již má řada routerů anténu vestavěnou. [42]

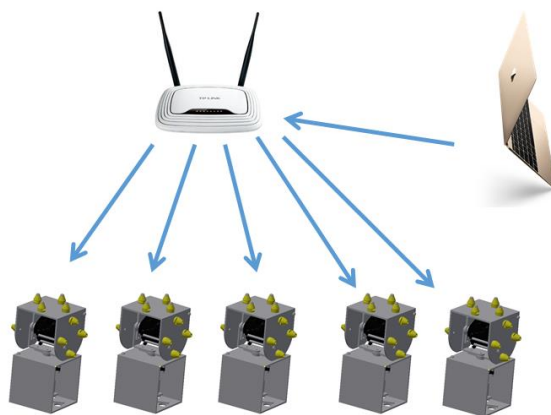
Rychlost Wi-fi je různá podle intenzity signálu (v zákoutích s nízkým signálem je přenosová rychlost nižší než na místech s plnou intenzitou) a dále je určena standardem. Nejstarší je IEEE 802.11b s maximální rychlostí 12Mbit/s. Mladší je IEEE 802.11g s maximální rychlostí 54Mbit/s a novější je IEEE 802.11n s přenosovou rychlostí až 600Mbit/s, nicméně většina zařízení pracuje s maximální rychlostí 150Mbit/s. Vývojová deska NodeMCU podporuje všechny tyto typy standardů. [42]

### 5.4 Možnosti připojení k Wi-fi modulu

Mikrokontrolér a jeho Wi-fi připojení je možno realizovat několika způsoby. Pro Molecube je nutné, aby bylo možné z notebooku ovládat několik modulů najednou. Díky tomu, že robot je ve vnitřním prostředí, tak je možné využít domácí Wi-fi router a připojit jednotlivé moduly k němu. Poté notebook, který je také připojen k routeru, odesílá požadavky jednotlivým modulům. Při připojování modulů k routeru je potřebné mít v programu uvedené jméno a heslo Wi-fi sítě, takže moduly mohou ovládat pouze uživatelé znající tyto informace. Druhou možností je komunikace jenom s jedním modulem bez využití routeru, pak je možné nahrát do mikrokontroléru program, který vytvoří vlastní přístupový bod. Mikrokontrolér pak vytvoří vlastní Wi-fi síť a k té je možné se připojit.



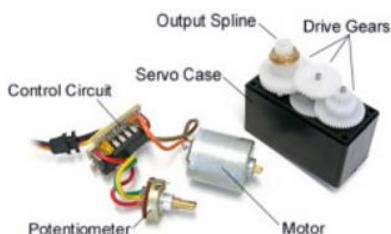
Obr. 45 Mód s vytvořením přístupového bodu [36]



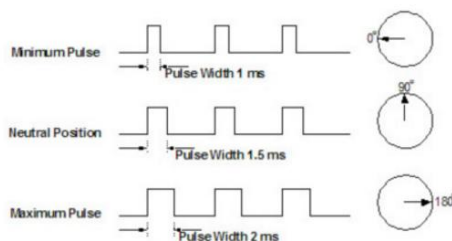
Obr. 46 Mód s využitím routeru [36][37]

## 5.5 Ovládání servomotorů

Hlavní součásti modelářského servomotoru jsou motor, převodová kola, kryt servomotoru, řídicí obvod a potenciometr, který poskytuje zpětnou vazbu o poloze (Obr. 47). Připojení servomotorů je pomocí třech kabelů. Dva kabely tvoří napájení a třetí kabel je pro řízení servomotoru. Do servomotoru se zasílá pulz každých 20 ms a podle délky tohoto pulzu motor nastavuje svoji pozici (Obr. 48). Pro správné fungování motoru je možné využít knihovnu, která je k dispozici ve vývojovém prostředí a jmenuje se *Servo.h*. [35][38]



Obr. 47 Komponenty servomotoru [38]



Obr. 48 Délka pulzu ovlivňuje natočení servomotoru [38]

Během připojení servomotorů je možné individuálně nastavovat rozsah pulzů každému servomotoru. K tomu je využit příkaz *servo attach (pin, min, max)* kde *pin* značí, kam je řídicí kabel servomotoru připojen, *min* je potom minimální délka pulzu pro 0° a *max* je maximální délka pulzu pro 180°. Obvyklá délka pro minimální a maximální hodnotu natočení činí 544 μs a 2400 μs. [35]

Program je vytvořený podle příkladu *Hello server*, který obsahuje připojení k routeru a reakci na požadavek klienta. [40]

Dále je zabudována reakce na požadavek, aby se motor nastavil na požadovanou hodnotu. Inspirací této tvorby byl program pro řízení servomotoru přes Wi-fi rozhraní. [38]

## 5.6 Program pro Molecube – server

Úkolem programu je připojit se k místní síti a poté zpracovávat požadavky klienta, který může požadovat buď natočení jednoho ze tří servomotorů, nebo zjištění výstupní hodnoty senzoru.

```
//program pro ovladani robota Molecube

/*****/
//inicializace

//pripojeni externich knihoven pro servomotory a Wi-fi
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

//informace o wi-fi routeru: nazev a heslo
const char* ssid = "*****";
const char* pass = "*****";

//urceni na kterem kanalu se bude komunikovat
ESP8266WebServer server(80);

//vytvoreni tri objektu typu servo
Servo horniServo;
Servo dolniServo;
Servo zajistovaciServo;
// Vytvoreni promenne pro uchovani hodnoty aktualniho natoceni
int NatoceniHorniServo = 90;
int NatoceniDolniServo = 90;

/*****/
//nastavovaci faze
void setup(void) {
    //zahajeni seriove komunikace, mozne ji zobrazit
    //po kliknuti na Nastroje/Seriovy monitor
    Serial.begin(115200);
    Serial.println("");

    //pripojeni k wi-fi routeru
    WiFi.begin(ssid, pass);
    // cekani na pripojeni
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    //po pripojeni vypsani ke ktere siti jsme pripojeni
    //take vypsani IP adresy modulu
```

```

Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

/*****/
//zpracovavani pozadavku

//pokud je za IP adresou napsane pouze lomitko
//tak vypis informace jak ovladat modul
server.on("/", []() {
    server.send(200, "text/plain", "Otevri:
    IP/(Horni) (Dolni) (Zajistovaci)Servo?value=90 pro nastaveni servomotoru, nebo
    /mikrospinac pro zjistení hodnoty senzoru");
});

//pokud je pozadavek na horniServo, nastav prislusnou hodnotu na servomotoru
server.on("/horniServo", []() {
    String sval = server.arg("value");
    int ival = sval.toInt();
    Serial.print("Servo: ");
    Serial.println(ival);
    //plynule nastaveni polohy
    if (ival>NatoceniHorniServo){
        for (int i=NatoceniHorniServo;i<=ival;i++){
            horniServo.write(i);
            delay(10);}
    }
    else {
        for (int i=NatoceniHorniServo;i>=ival;i--){
            horniServo.write(i);
            delay(10);}
    }

    server.send(200, "text/plain", String(ival, DEC));
});

//pokud je pozadavek na dolniServo, nastav prislusnou hodnotu na servomotoru
server.on("/dolniServo", []() {
    String sval2 = server.arg("value");
    int ival2 = sval2.toInt();
    Serial.print("Servo: ");
    Serial.println(ival2);
    //plynule nastaveni polohy
    if (ival2>NatoceniDolniServo){
        for (int i=NatoceniDolniServo;i<=ival2;i++){
            horniServo.write(i);
            delay(10);}
    }
    else {

```

```

    for (int i=NatoceniDolniServo;i>=ival2;i--){
        horniServo.write(i);
        delay(10);}
    }

    server.send(200, "text/plain", String(ival2, DEC));
});

//pokud je pozadavek na zajistovaciServo, nastav prislusnou hodnotu na servomotoru
server.on("/zajistovaciServo", []() {
    String sval3 = server.arg("value");
    int ival3 = sval3.toInt();
    Serial.print("Servo: ");
    Serial.println(ival3);
    zajistovaciServo.write(ival3);
    server.send(200, "text/plain", String(ival3, DEC));
});

//pokud je pozadavek na mikrospinac, precist vstup a odeslat vysledek
server.on("/mikrospinac", []() {
    Serial.print("Mikrospinac: ");
    int hodnota = digitalRead(00);
    Serial.println(hodnota);
    server.send(200, "text/plain", String(hodnota, DEC));
});

server.begin();
//vypis na seriový monitor hlášení o spuštění serveru
Serial.println("HTTP server started");

//připojení jednotlivých servomotorů na příslušné piny
//nastavení polohy každého servomotoru po zapnutí na 90 stupňů
horniServo.attach(16); // Servo připojeno k D0 pinu na desce NodeMCU
horniServo.write(90);
dolniServo.attach(05); // Servo připojeno k D1 pinu na desce NodeMCU
dolniServo.write(90);
zajistovaciServo.attach(04); // Servo připojeno k D2 pinu na desce NodeMCU
zajistovaciServo.write(90);

//aktivace pinu D3 na desce NodeMCU jako vstupního pinu z mikrospinace
pinMode(00, INPUT_PULLUP);
}

/*****/
//hlavní smyčka
void loop(void) {
    //hlavní smyčka probíhá stále dokola
    //neustále zpracováváme klientovy požadavky
    server.handleClient();
}

```

## 5.7 Program pro Molecube – přístupový bod

Program je obdobný, pouze jsou změny v inicializační fázi a částečně v nastavovací fázi – místo připojování k routeru se vytváří vlastní přístupový bod. K tomu je také zapotřebí připojení další knihovny. Zde je uvedena pouze inicializační a část nastavovací fáze: [41]

```
//program pro ovladani robota Molecube

/*****/

//inicializace

//pripojeni externich knihoven pro servomotory a Wi-fi
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

//vytvoreni vlastni Wi-fi: volba nazvu a hesla
const char* ssid = "MOLECUBE";
const char* password = "1234";

//urceni na kterem kanalu se bude komunikovat
ESP8266WebServer server(80);

/* Pro overeni funkcnosti zadat http://192.168.4.1 do * webového prohlizece */
void handleRoot() {
    server.send(200, "text/html", "<h1>Jste pripojeni</h1>");
}

//vytvoreni tri objektu typu servo
Servo horniServo;
Servo dolniServo;
Servo zajistovaciServo;

// Vytvoreni promenne pro uchovani hodnoty aktualniho natoceni
int NatoceniHorniServo = 90;
int NatoceniDolniServo = 90;

/*****/

//nastavovaci faze
void setup(void) {
    //zahajeni seriove komunikace, mozne ji zobrazit
    //po kliknuti na Nastroje/Seriový monitor
    Serial.begin(115200);
    Serial.println("");

    /* * Vytvoreni Wi-fi pristupoveho bodu */
```

```
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
//po dokonceni vypsani informaci o IP adrese do serioveho monitoru
Serial.print("AP IP address: ");
Serial.println(myIP);
server.on("/", handleRoot);
server.begin();
//vypsani hlaseni o spusteni serveru
Serial.println("HTTP server started");

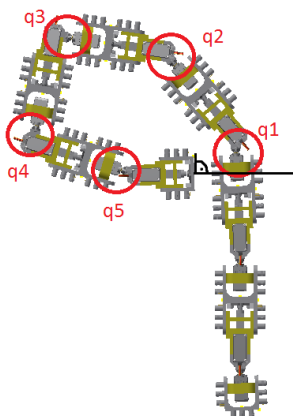
/*****
```



## 6 PŘÍMÁ A INVERZNÍ ÚLOHA KINEMATIKY

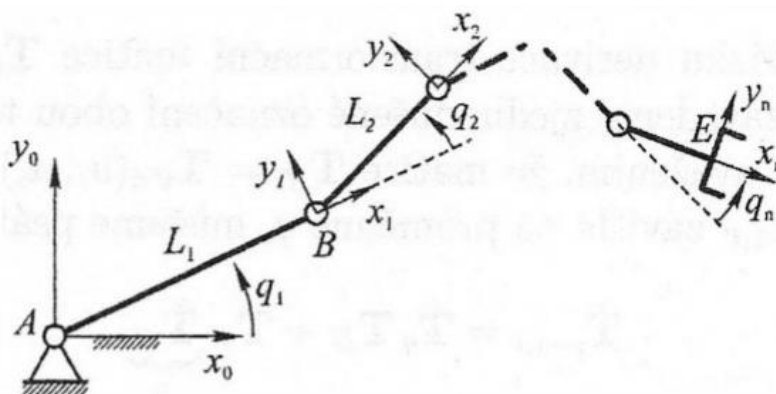
### 6.1 Využití přímé a inverzní kinematiky pro rekonfiguraci robota

Během rekonfigurace potřebujeme znát, jak musí robot jednotlivé moduly natočit, aby mohlo dojít k novému propojení a tak ke změně struktury robota. Této úloze, kdy hledáme natočení  $q$  potřebné pro dosažení cílové pozice, se věnuje inverzní kinematika, která pro svůj výpočet používá přímou kinematiku.



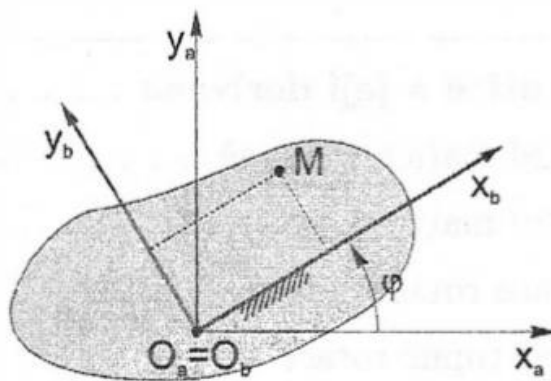
Obr. 49 Pro propojení dvou modulů musíme znát úhly natočení

### 6.2 Přímá úloha kinematiky



Obr. 50 Otevřený kinematický řetězec [4]

Při řešení přímé úlohy kinematiky známe jednotlivá natočení nebo posunutí v kloubech (na obrázku jsou vyznačené kolečky (Obr. 50)) a známe rozměry manipulátoru. Hledáme polohu a orientaci koncového efektoru v kartézském souřadném systému. Pro tento výpočet se využívají rotační matice. Rotační matice přepočítává souřadnice dvou vzájemně pootočených souřadných systémů. Rotace v rovině je znázorněná na obrázku (Obr. 51). [4]



Obr. 51 Rotace dvou souřadnicových systémů [4]

Rotační matici v rovině je možné odvodit z výpočtu (6.1) a (6.2): [4]

$$x_a = \cos\varphi x_b - \sin\varphi y_b \quad (6.1)$$

$$y_a = \sin\varphi x_b + \cos\varphi y_b \quad (6.2)$$

Po přepsání do maticového tvaru má rovnice tvar (6.3).  $\mathbf{R}_{ba}$  je rotační matice  $R$  ze systému  $b$  do systému  $a$ . [4]

$$\mathbf{r}_a^M = \begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_b \\ y_b \end{bmatrix} = \mathbf{R}_{ba} \mathbf{r}_b^M \quad (6.3)$$

V prostoru je rotační matice podstatně složitější, neboť orientace tělesa již není popsána pouze jedním úhlem, ale třemi úhly. Nicméně existují speciální tvary rotační matice pro rotaci kolem jednotlivých os. Některé prvky v těchto maticích jsou nenulové. [4]

Rotace systému  $b$  okolo osy  $x$ : [4]

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix} \quad (6.4)$$

Rotace systému  $b$  okolo osy  $y$ : [4]

$$R_y = \begin{bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \quad (6.5)$$

Rotace systému  $b$  okolo osy  $z$ : [4]

$$R_z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.6)$$

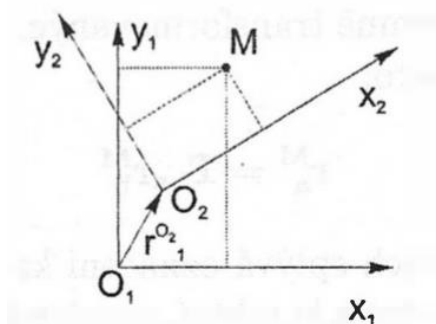
### 6.2.1 Dvě postupné rotace

Celkovou rotaci lze vyjádřit pomocí dílčích rotací násobením jednotlivých rotačních matic, přičemž záleží na pořadí násobených matic. Systém  $c$  je pevně spojený s tělesem, systém  $b$  je vůči němu natočen a systém  $a$  je natočený vůči  $b$ , přičemž  $a$  reprezentuje např. pevný rám. Výslednou rotaci potom získáme: [4]

$$R_{ca} = R_{ba} \cdot R_{cb} \quad (6.7)$$

### 6.2.2 Transformační matice

Rotaci i translaci tělesa v prostoru popisuje transformační matice. Posun a natočení ve 2D prostoru je vykresleno na obrázku (Obr. 52). Maticový zápis je potom tvořen rovnicí (6.8). [4]



Obr. 52 Posun a natočení dvou souřadnicových systémů v rovině [4]

$$\begin{bmatrix} x_a^M \\ y_a^M \\ z_a^M \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & x_a^{O_b} \\ \sin\varphi & \cos\varphi & 0 & y_a^{O_b} \\ 0 & 0 & 1 & z_a^{O_b} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_b^M \\ y_b^M \\ z_b^M \\ 1 \end{bmatrix} \quad (6.8)$$

### 6.3 Inverzní úloha kinematiky

Popisuje úlohu, kdy známe polohu a orientaci koncového efektoru v kartézském souřadném systému, známe také rozměry mechanismu a hledáme kloubové souřadnice. Přímoúhlohu kinematiky je možné řešit v uzavřeném tvaru, oproti tomu inverzní kinematika vede na řešení silně nelineárních algebraických rovnic, které je ve většině případů nutno vypočítat iteračními numerickými metodami s využitím Jakobiánu. [4]

Matematické obtíže při řešení: [4]

- Omezený pracovní prostor – manipulátor pracuje v omezeném prostoru – polohy a orientace mimo tento prostor neexistují
- Víceznačnost řešení - jedna poloha může být dosažena více způsoby
- Redundantní manipulátor – nadbytečný počet kloubových souřadnic (délka vektoru  $q$  je větší než počet požadovaných souřadnic (v prostoru max. 6)) - požadovanou polohu pak můžeme dosáhnout nekonečně mnoha způsoby
- Singulární stav manipulátoru, kdy vznikají numerické obtíže při výpočtu – determinant Jakobiánu je roven nule. Pseudoinverzí nebo transpozicí lze ze singulárního stavu uniknout. U redundantního manipulátoru při singulárním stavu dojde ke snížení hodnosti matice (snížení počtu nezávislých řádků nebo sloupců matice).

Pro numerické řešení převedeme totální diferenciál na difference: [4]

$$\begin{aligned}\Delta x &= J \Delta q \\ \Delta q &= J^{-1} \Delta x\end{aligned}\tag{6.9}$$

Iterační proces hledání kloubových souřadnic potom probíhá podle vzorce: [4]

$$\begin{aligned}x_k &= f(q_k) \\ q_{k+1} &= q_k + J^{-1}(\tilde{x} - x_k)\end{aligned}\tag{6.10}$$

Narážíme však na několik omezení - determinant se nesmí rovnat nule a matice Jakobiánu musí být čtvercová. Obě tato omezení řeší Pseudoinverze, Transpozice a Metoda tlumených nejmenších čtverců. [4]

### 6.3.1 Pseudoinverze

Klasická inverze je nahrazena pseudoinverzí a umožní tak vyřešit redundantní manipulátor. [4]

$$\begin{aligned}\Delta q &= J^+ \Delta x \\ J^+ &= J^T (JJ^T)^{-1}\end{aligned}\tag{6.11}$$

### 6.3.2 Transpozice

Využití transpozice místo inverze a přidáme parametr alfa: [4]

$$\Delta q = \alpha J^T \Delta x\tag{6.12}$$

### 6.3.3 Metoda tlumených nejmenších čtverců

Metoda Levenberg-Marquardtova hledá takové  $\Delta q$ , které minimalizuje výraz: [4]

$$\|J \Delta q - \Delta x\|^2 + \lambda \|\Delta q\|^2\tag{6.13}$$

Kde  $\lambda$  je tlumící konstanta, jejíž velikost musí být dostatečná pro vyřešení problémů kolem singulárních stavů, a naopak dostatečně malá kvůli rychlé konvergenci. [4]

Z rovnice lze vyjádřit výpočet natočení kloubů: [4]

$$\Delta q = J^T (JJ^T + \lambda^2 I)^{-1} \Delta x\tag{6.14}$$

Analytický výpočet Jakobiánu - je vypočítán jako matice parciálních derivací: [4]

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \cdots & \frac{\partial f_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix}\tag{6.15}$$

## 6.4 Aplikace numerické metody

Pro výpočet natočení modulů byla využita modifikovaná metoda tlumení nejmenších čtverců. Umožňuje nejen řešit pře určené systémy, což robot Molecube je, ale je tu i možnost zvolení si více bodů, kterými má robot procházet a na zvolené body dát důraz podle parametru lambda. [4]

Algoritmus výpočtu inverzní kinematiky: [4]

```

λ =diag(λ1,... , λm)           %zvolení si jednotlivých parametrů lambda
[X]=fkine[Q]                     %výpočet dopředné kinematiky
While err>0.01                   %smyčku provádět, dokud není chyba malá
    J=Jacobian[Q]                %výpočet matice Jakobiánu
    dX=(XG-X)                    %jak daleko jsme vzdáleni od cíle
    dQ=JT*(J*JT+ λ)-1*dX      %jak natočit jednotlivé klouby
    Q=Q+dQ                       %aktualizace hodnot natočení kloubů
    X=fkine(Q)                   %výpočet dopředné kinematiky
    Err=norm(X-XG)               %Err = délka vektoru dX
end

```

## 6.5 Využití parametru lambda

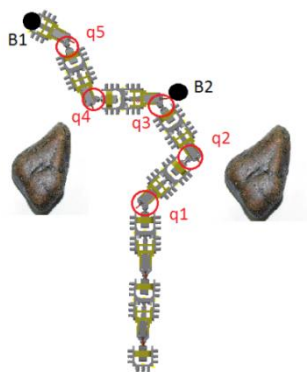
Tato možnost metody tlumených nejmenších čtverců se velmi hodí, protože pokud definuji koncový bod, který má robot dosáhnout, mohu definovat i další bod, kterým musí manipulátor také procházet. Také je možné zadat požadované natočení kteréhokoliv kloubu a algoritmus se pak náš požadavek snaží dodržet. Díky tomu je možné vyhýbat se překážkám nebo předpokládat, v jaké poloze se robot asi bude nacházet. Tato metoda také zohledňuje, jak moc je pro nás který bod důležitý – díky hodnotě parametru lambda. [4]

Jak velikost parametru lambda ovlivňuje výsledky je ukázáno na následujících obrázcích. Úloha je následující: Robot je mezi překážkami, které jsou tvořeny dvěma kameny, na grafech jsou znázorněny jako dvě modré hvězdičky. Má za úkol dosáhnout do bodu B1 za předpokladu, že musí procházet i bodem B2. To nám zajistí, že nenarazí do žádné překážky a můžeme tedy předpovídat, jak bude výsledek vypadat. (Obr. 53)

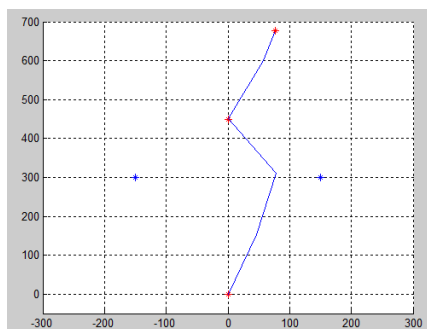
Pokud zvolíme parametr lambda stejný pro oba body, bude výsledek vypadat jako na obrázku (

Obr. 54). Na obou bodech nám záleží stejně a chyba přesnosti v obou bodech bude tedy přibližně stejná. Pokud upravíme hodnotu lambda tak, aby záleželo více na cílovém bodě (

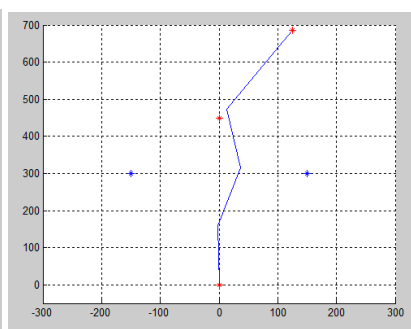
Obr. 55), dosáhneme toho, že cílový bod bude s velkou přesností dosažen, ovšem bod B2 je zohledněn méně. Toho můžeme využít, pokud robotem chceme manipulovat za překážkou (Obr. 56). V tomto případě robot dosáhne požadovaného bodu B1. Bod B2 je sice s menší přesností, ale to není problém, protože zde jde hlavně o to, aby nenastala kolize s překážkou a to se podařilo.



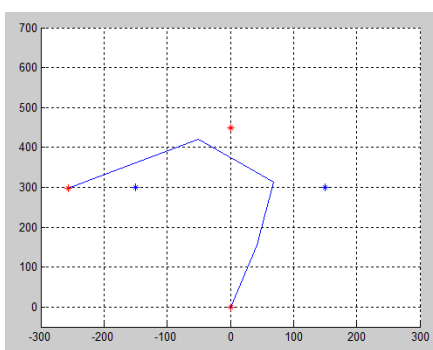
Obr. 53 Manipulace mezi překážkami



Obr. 54 Hodnoty parametru lambda jsou stejné



Obr. 55 Úprava hodnot lambda tak, aby záleželo více na cílovém bodě



Obr. 56 Manipulace za překážkou

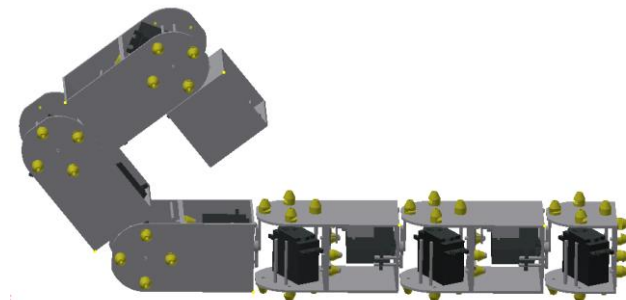
## 7 REKONFIGURACE ROBOTA

### 7.1 Průběh rekonfigurace

Během rekonfigurace se moduly natočí tak, aby koncový modul své čtyři otvory přesně nasunul na přípojovací kužely umístěné na boku jiného modulu. Poté je nově vzniklé spojení zajištěno zajišťovacím křížem. V jiném místě se moduly rozpojí. Potom se moduly natočí do základní polohy ( $90^\circ$ ) a je patrná nová struktura robota. Díky výpočtům inverzní kinematiky je možné si potřebná natočení jednotlivých servomotorů pro rekonfiguraci vypočítat.

### 7.2 Požadavky na rekonfiguraci

Každý modul má na spodní straně otvory se zajišťovacím křížem pro připojení k přípojovacím kuželům. Tuto stranu je možné nazvat aktivním konektorem. V sériovém spojení modulů za sebou je patrné, že robot má volný pouze jeden aktivní konektor. Tento konektor můžeme zapojit do ostatních modulů. Každý modul při sériovém zapojení má ještě dvě volné strany s přípojovacími kužely, kromě posledního členu, ten má možné přípojovací strany tři (Obr. 57). Úlohu je postačující navrhnout pouze ve 2D rovině, kdy se výpočet zjednoduší oproti 3D prostoru. Toho je dosaženo díky následující skutečnosti – natočením propojovaného modulu o  $90$  stupňů se úloha mění ze 3D úlohy do 2D úlohy (Obr. 57). Rekonfigurace tedy bude probíhat tak, že se moduly mohou propojovat nad sebou (ve vertikální rovině), nebo jako ležící na boku (v horizontální rovině).



Obr. 57 Natočení propojovaného modulu

Pokud je vybrán modul, se kterým se má aktivní konec spojit, je možné vypočítat, kolik ramen má provádět natáčení a vypočítat inverzní výpočet. Další podmínkou pro správnou funkčnost je, aby se moduly do sebe zasouvaly kolmo.

### 7.3 Vstup a výstup řešení

Vstupem do funkce je požadavek, které moduly se mají propojit. Výstupem bude grafické znázornění, jak bude výsledná varianta vypadat a poté možnost odeslat vypočítané úhly jednotlivým servomotorům. Tyto úhly musí být ve stupních a zaokrouhlené na celé čísla, aby hodnota mohla být zpracována servomotory. Natočení jednotlivých servomotorů v modulech pak řídí mikrokontrolér, na kterém je nahrána příslušná knihovna pro ovládání servomotorů.

## 7.4 Odhad řešení

Velkým problémem se ale jeví, že mnoho řešení, které spočítám, budou sice matematicky správně, je ale požadováno natočení všech kloubů pouze v rozmezí 0-180°. Pokud jsou vypočítány jiné úhly, musí výpočet proběhnout znovu. Proto bude pravděpodobně nutné tuto podmínku ve výpočtu nějakým způsobem zohlednit.

## 7.5 Funkce pro výpočet natočení

Podle zadaných podmínek je vytvořená následující funkce, která na základě zadaného cílového bodu ve 2D rovině a požadované orientaci koncového členu manipulátoru vypočítá natočení jednotlivých kloubů. Zadán je tedy vektor  $\mathbf{xG}$  který obsahuje požadovanou polohu  $x$ , polohu  $y$  a natočení koncového efektoru. Zadány jsou také rozměry manipulátoru ve vektoru  $L$ . Tento vektor obsahuje délky ramen v pořadí od první osy rotace, tedy délka koncového efektoru je jako poslední. Funkce poté vrátí jednotlivá natočení kloubů.

Pro větší přehlednost je výpočet navržen tak, aby matice Jakobiánu a dopředná kinematika byla jako oddělená funkce, kterou funkce pro výpočet natočení využívá.

```
%inverzni kinematika
function [ q ] = algoritmus(xG,L)
% pocateni uhly natoceni
q=zeros(length(L),1);
q(:)=0;

% inicializace promenne err
err=10;
% vypocet dopredne kinematiky
[x]=dopred_kinematika(q,L);
%volby parametru lambda, pro x, y a natoceni
lambda=[1 0 0;0 1 0; 0 0 1];
%kontrola proti precykleni
i=0;
%iteracni smycka pro vypocet inverzni kinematiky
while err>0.01
[J] = jacobian(L,q);
dx=(xG-x);
dampedJ=(J.')*(J*(J.')+lambda)^-1;
dq=dampedJ*dx;
q=q+dq;

%kontrola proti precykleni
i=i+1;
```



```

if i>1000
    break;
end

[x]=dopred_kinematika(q,L);
err=norm(x-xG);

end

```

## 7.6 Funkce pro dopřednou kinematiku

Pro výpočet potřebné dopředné kinematiky je vytvořena funkce, do které vstupují úhly natočení a jednotlivé délky ramen. Výstupem z funkce je poloha koncového efektoru a natočení koncového efektoru. Funkce je schopná přijímat až 5 úhlů natočení a podle počtu členů zvolí vhodný výpočet. Jednotlivé matice se vypočítaly násobením matic transpozice obsahující rotaci kolem osy z. Uvedeny jsou zde první dvě vypočítané transponované matice. První matice (*case 1*) obsahuje pouze jedno natočení a jedno posunutí. Další matice (*case 2*) již řeší polohu koncového bodu po dvou rotacích a dvou posunutích. Hodnota, kterou si z matice vybíráme je poloha bodu  $x$ , která je v matici na prvním řádku ve čtvrtém sloupci. Dále nás zajímá hodnota  $y$ , ta je na druhém řádku ve čtvrtém sloupci. Poslední výstupné hodnotou je celkové natočení  $q$ , které je vypočítáno jako součet jednotlivých natočení.

Funkce pro dopřednou kinematiku:

```

function [ x ] = dopred_kinematika( q,L )
switch length(L)
    case 1
        T=[cos(q(1)) -sin(q(1)) 0 L(1)*cos(q(1));
           sin(q(1)) cos(q(1)) 0 L(1)*sin(q(1));
           0 0 1 0;
           0 0 0 1];
        %natoceni koncoveho efektoru
        x(3,1)=q(1);
    case 2
        T=[cos(q(1)+q(2)) -sin(q(1)+q(2)) 0 L(2)*cos(q(1)+q(2))+L(1)*cos(q(1));
           sin(q(1)+q(2)) cos(q(1)+q(2)) 0 L(2)*sin(q(1)+q(2))+L(1)*sin(q(1));
           0 0 1 0;
           0 0 0 1];
        %natoceni koncoveho efektoru
        x(3,1)=q(1)+q(2);
end

```

```

x(1,1)=T(1,4);
x(2,1)=T(2,4);

end

```

## 7.7 Funkce pro Jakobiho matici

Jakobiho matice koresponduje s přímou kinematikou. Tedy na základě počtu otáčejících se ramen je vybrána příslušná už připravená Jakobiho matice. Ta je vypočítána parciální derivací z přímé úlohy kinematiky. Počet otáčejících se ramen ovlivňuje velikost Jakobiho matice. Pokud se jedná o nejjednodušší problém (pouze jednu rotaci), je výsledná Jakobiho matice v části *case 1*. První řádek Jakobiho matice je parciální derivace polohy  $x$ , která se nachází v matici polohy na prvním řádku a ve čtvrtém sloupci:  $L(1)*\cos(q(1))$ , derivace podle  $q(1)$  je tedy:  $-L(1)*\sin(q(1))$  jak je vidět na výsledku. Další řádek v Jakobiho matici je derivace polohy  $y$  podle  $q(1)$  a poslední řádek je natočení koncového efektoru. Toto natočení má hodnotu  $q(1)$ , derivací podle  $q(1)$  získáváme tedy hodnotu 1. Jakobiho matici popisuje vztah (6.15). Pro více natočení je patrné, že matice získává obdélníkový tvar, což je charakteristické pro redundantní manipulátor. Uvedeny jsou první tři výpočty Jakobiho matice pro jednu rotaci (*case 1*), dvě rotace (*case 2*) a tři rotace (*case 3*).

Funkce pro výpočet Jakobiho matice:

```

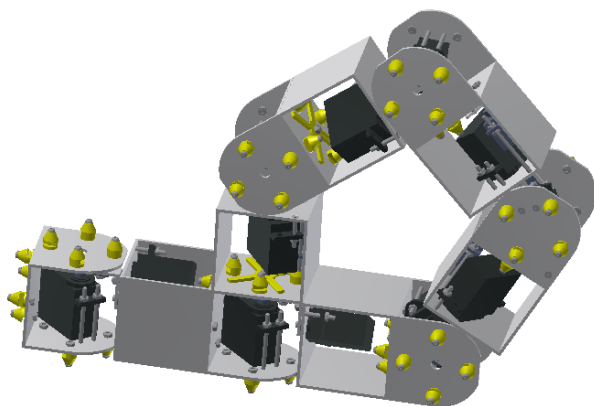
function [ J ] = jacobian( L,q )
switch length(L)
case 1
    J=[-L(1)*sin(q(1));
        L(1)*cos(q(1));
        1];
case 2
    J=[-L(2)*sin(q(1)+q(2))-L(1)*sin(q(1)) -L(2)*sin(q(1)+q(2));
        L(2)*cos(q(1)+q(2))+L(1)*cos(q(1)) L(2)*cos(q(1)+q(2));
        1 1];
case 3
    J=[-L(3)*sin(q(1)+q(2)+q(3))-L(2)*sin(q(1)+q(2))-L(1)*sin(q(1))
        -L(3)*sin(q(1)+q(2)+q(3))-L(2)*sin(q(1)+q(2)) -L(3)*sin(q(1)+q(2)+q(3));
        L(3)*cos(q(1)+q(2)+q(3))+L(2)*cos(q(1)+q(2))+L(1)*cos(q(1))
        L(3)*cos(q(1)+q(2)+q(3))+L(2)*cos(q(1)+q(2)) L(3)*cos(q(1)+q(2)+q(3));
        1 1 1];
end

```

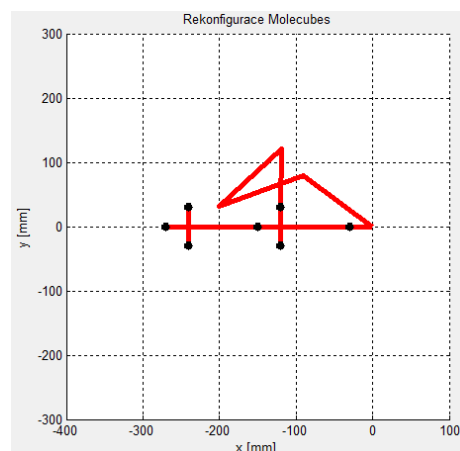
## 7.8 Omezení rozsahu pohybu v kloubu

Byl zvolen požadavek na rekonfiguraci, aby se modul koncovým efektozem připojil do boku pátého modulu (Obr. 58). Během výpočtu a vykreslení řešení je patrný předpokládaný

problém - konvergence je sice poměrně rychlá, požadovaná přesnost dosažení koncového bodu  $[-120;30]$  je už po dvanácti iteracích, nicméně nalezené řešení není vhodné (Obr. 59), protože modul má omezený rozsah natočení  $-90$  až  $+90$  stupňů.



Obr. 58 Požadovaná rekonfigurace



Obr. 59 Nevhodné řešení rekonfigurace

Nyní máme dvě možnosti, jak se dopracovat k jinému výsledku:

- 1) Změna počátečních podmínek natočení:
  - Navržení algoritmu, který zkontroluje rozsah jednotlivých  $q$  natočení a pokud nejsou v rozmezí  $-\pi/2$  až  $+\pi/2$  radiánů, tak změni počáteční podmínku (odhad natočení) a výpočet se provede znovu. Výhodou této metody je jednoduchost, neboť není nutno zasahovat do algoritmu inverzní kinematiky, nevýhodou je, že nalezení vhodných bodů je takto poměrně náhodné a nesystematické, nebo také že se nepřibližujeme k výslednému řešení postupně, ale pouze jsou zkoušena různá čísla dosazovat do výpočtu a čeká se, kdy vyjde vhodné řešení. Toto řešení není vyhovující.
- 2) Během výpočtu zasahovat do algoritmu:
  - Když vyjde natočení v kloubu a je mimo povolený rozsah, tak upravit hodnotu zpět do rozsahu. Chyba výsledného vektoru sice potom vyjde větší a je tak mírně narušena stabilita výpočtu, nicméně je poměrně rychle a systematicky přibližováno ke vhodnému řešení.

Druhé řešení je z hlediska systematiky a rychlosti mnohem lepší, proto byla do algoritmu vložena podmínka, která během každé iterace porovná rozsah natočení úhlů a pokud je navržený úhel větší než povolený rozsah, tak hodnotu upraví zpět. Výsledná konvergence je po 14ti iteracích, což je jen velmi malé zpomalení a výsledek je již pro rekonfiguraci použitelný (Obr. 60). Na obrázku je také možné vidět, jak probíhají jednotlivé iterační výpočty, než je nalezeno vhodné řešení. Je vidět, jak se postupně k řešení výpočet přibližuje i jak není nikdy překročen omezený rozsah natočení (Obr. 61).

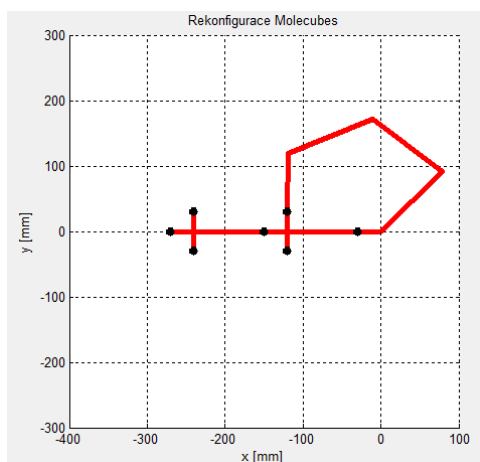
Podmínka omezující maximální rozsah natočení v kloubech:

```
if q(e) < -pi/2
    q(e) = -pi/2;
end
```

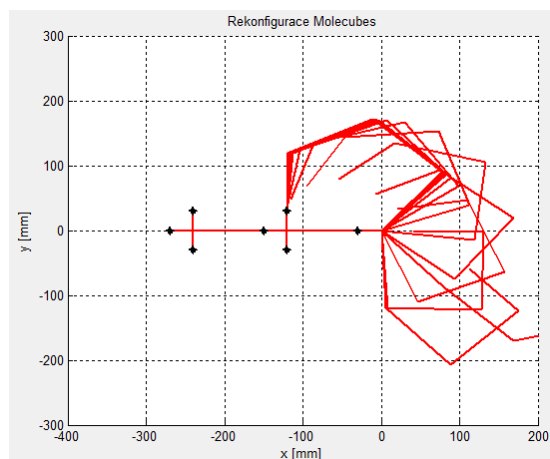
```

if q(e)>pi/2
    q(e)=pi/2;
end

```



Obr. 60 Rekonfigurace po omezení natočení v kloubech



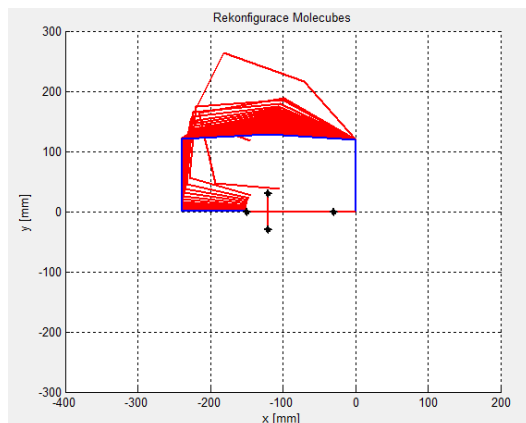
Obr. 61 Jednotlivá řešení během výpočtu

## 7.9 Urychlení konvergence pro triviální případy

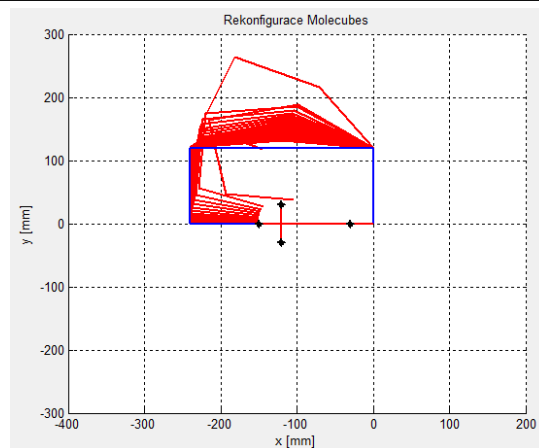
Během testování vyplynulo, že pokud se jedná o výpočet transformace, kdy požadujeme řešení nějakého triviálního úkolu (jako např. na obrázku, kdy jsou natočení buď 0, nebo 90 stupňů), tak výpočet není schopen dosáhnout požadované přesnosti řešení ani po 1000 iteracích (Obr. 62). Poslední iterace je naznačena modře. Jedná se o rekonfiguraci vhodnou pro lokomoci, kdy jsou všechny moduly propojeny vzájemně dokola aby se mohly ovalovat.

Tato úloha by mohla být vyřešena tím, že pokud výpočet stále nemůže nalézt vhodné řešení, tak až uplyne 500 iterace, tak algoritmus zhodnotí, jestli se úhel některého natočení neblíží nule (buď zprava, nebo zleva) a pokud ano, tak nahradí hodnotu nulou. Dále pokud se blíží hodnota natočení  $+\pi/2$  nebo  $-\pi/2$ , nahradí stávající hodnotu touto přesnou hodnotou.

Výsledek je možné pozorovat na obrázku, kdy nalezení vhodného řešení je okamžité po dosažení 500 iterací, kdy dojde k nahrazení přibližné hodnoty natočení přesnou hodnotou. Poslední iterace je naznačena modře a je patrné, že se jedná o požadované řešení.



Obr. 62 Neschopnost nelézt přesné řešení



Obr. 63 Konvergence nastane ihned po překročení 500 iterací

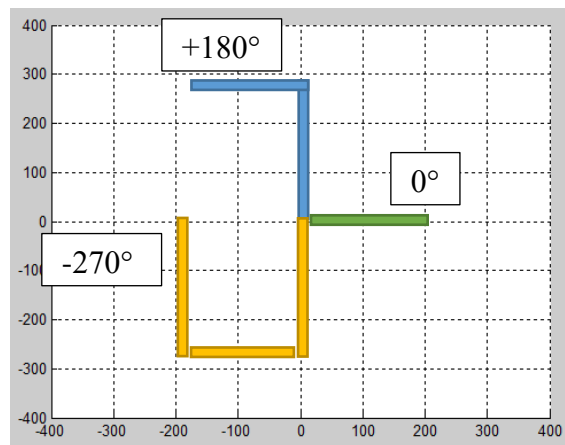
Ukázka části programu, který je vložen do algoritmu a při překročení 500 iterací provádí porovnání hodnot:

```

if i>500
    for f=1:length(L)
        if q(f)<0.2&&q(f)>-0.2
            q(f)=0;
        end
        if q(f)>1.3
            q(f)=pi/2;
        end
        if q(f)<-1.3
            q(f)=-pi/2;
        end
    end
end
    
```

## 7.10 Orientace koncového efektoru

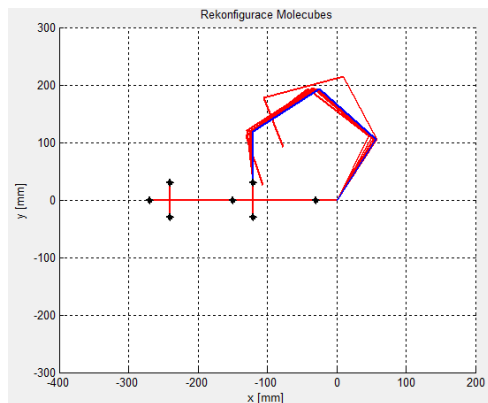
Pro správnou práci s programem Matlab je nutné znát, jak zadat požadovanou orientaci koncového efektoru. Ta je pro nás důležitá, neboť moduly se do sebe musí zasouvat kolmo. Matlab provádí výpočty v radiánech. Nulový úhel leží na vodorovné ose směrem doprava. Při natáčení směrem doleva nad bodem [0;0] se výsledný úhel sčítá a je kladný. Hodnota výsledného úhlu  $+180^\circ$  a  $-270^\circ$  je na obrázku (Obr. 64). Pokud je požadavek propojit moduly dokola pro lokomoci, výsledný úhel je 360 stupňů, neboli  $2\pi$  radiánů.



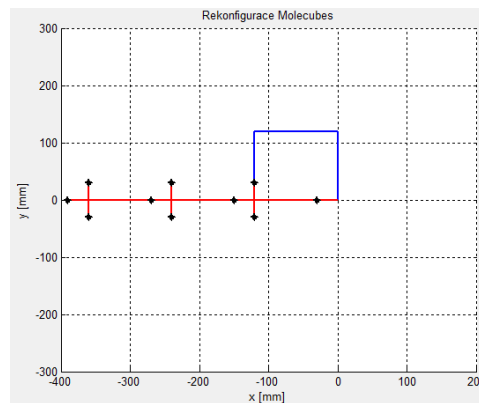
Obr. 64 Úhel natočení koncového efektoru

## 7.11 Počáteční podmínky

Bylo zjištěno, že pro správnou konvergenci záleží poměrně hodně na počátečních podmínkách. Nejpatrnější je to u manipulátoru se třemi rameny, které se mají ohnout do pravého úhlu. Tam konvergence při poč. podmínce, kdy jsou všechny úhly nulové, nebyla vůbec žádná. Toto zjištění vede k zavedení podmínky ve funkci inverzní kinematiky, která kontroluje, jestli chceme koncový modul připojit do levé strany pátého modulu (v grafu se toto jeví jako propojení v horní části), nebo do pravé strany pátého modulu (v grafu to vypadá jako propojení ve spodní části). Poté se nastaví pro horní propojení  $+\pi/2$  jako počáteční podmínku natočení pro všechny úhly a pro dolní propojení  $-\pi/2$  jako počáteční volbu natočení. Výsledek je možné pozorovat na obrázku (Obr. 65). Konvergence je mnohem rychlejší a je možné realizovat i jednoduchá triviální řešení, kdy je konvergence nalezena díky vhodné poč. podmínce hned v první iteraci (Obr. 66).



Obr. 65 Konvergence po upravených poč. podmínkách



Obr. 66 Okamžité nalezení řešení

Ukázka výpisu programu, kde se zajímáme o třetí hodnotu ve vektoru  $\mathbf{xG}$ , což je natočení koncového efektoru. Pokud je požadovaná hodnota větší než nula, navrhujeme kladné natočení pro všechny klouby, pokud je menší než nula, navrhujeme záporné hodnoty natočení:

```
if xG(3) > 0
    q(:) = pi/2;
elseif xG(3) < 0
    q(:) = -pi/2;
```

```
else
    q(:)=0;
end
```

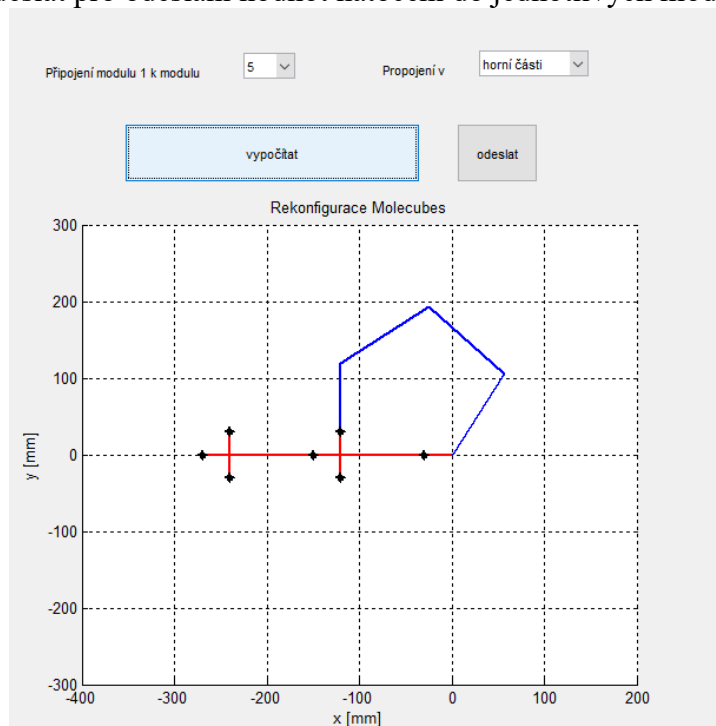
## 7.12 Grafické uživatelské rozhraní

Pro využití rekonfigurace bylo vytvořeno jednoduché uživatelské rozhraní, kde si uživatel může zvolit, do kterého modulu se má koncový modul připojit. Toto rozhraní vzniklo na základě těchto skutečností:

- Jen jeden konec modulu je aktivní, to znamená, že pracovat můžeme právě jen s tímto koncem
- Minimální počet modulů pro rekonfiguraci jsou 4, je to dáno omezením natočení jednotlivých modulů a jejich konstrukcí, nabízet tedy možnost propojení pro menší počet nemá smysl
- Rozměry jednotlivých modulů jsou známé, lze tedy z nich odvodit ramena manipulátorů

Uživatelské rozhraní se skládá z několika ovládacích prvků: (Obr. 67)

- Výběru, ke kterému modulu se má aktivní modul připojit
- Výběru, ke které straně modulu se má modul připojit (každý modul má tři připojovací strany)
- Tlačítko vypočítat pro spuštění výpočtu
- Tlačítko odeslat pro odeslání hodnot natočení do jednotlivých modulů



Obr. 67 Grafické uživatelské rozhraní

### 7.13 Odeslání http příkazu z Matlabu

V následující funkci je ukázána možnost odeslání *http* příkazu přímo z Matlabu. Takto je možné ovládat jednotlivé moduly přímo z programu Matlab. Odesílaný příkaz se skládá z IP adresy, která je každému Molecube modulu automaticky přidělená a z požadavku na Molecube. Ve funkci je využit požadavek na *horniServo*, což je horní servomotor. Tento servomotor přijímá požadavky na natočení.

```
function [ ] = OdeslaniNatoceni( )  
  
load Q.mat  
  
for i=1:length(Q)  
    data = sprintf('http://192.168.4.1/horniServo?value=%d%',Q(i));  
    s = urlread(data);  
  
end  
  
end
```

Mezi odesílanými časy je vhodné zařadit časové prodlevy, pro ně se dá využít příkaz pro pauzu – *pause(1)*. Výsledné odesílané hodnoty musí být převedeny z radiánů na stupně a zaokrouhleny, k tomu slouží příkaz  $Q=\text{round}(q*180/\pi)$ .

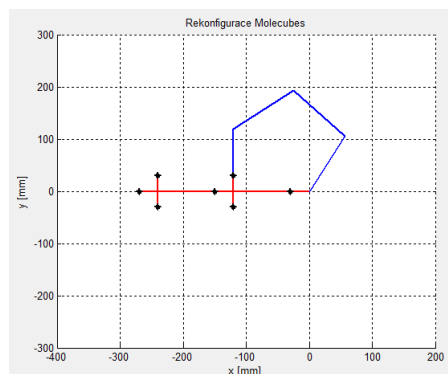
### 7.14 Testování rekonfigurace

#### 7.14.1 Podmínky testování

Všech šest modulů bylo vyrobeno shodně z konstrukčního z hlediska. Pro napájení a řízení se bohužel nepodařilo sehnat všechny komponenty pro všech šest modulů, proto byl během testu použit pro řízení některých modulů mikrokontrolér Arduino a některé moduly byly napájeny externě. Díky této modifikaci mohl test proběhnout. Vzhledem k tomu, že hmotnosti modulů jsou téměř totožné jako navržených modulů a řídicí mikrokontrolér využívá stejné knihovny na ovládání servomotorů, výsledky testu by měly být stejné.

#### 7.14.2 Testování propojení v horizontální rovině

Výchozí stav je, že jsou všechny moduly zapojeny sériově za sebou. Nyní je požadováno připojení koncového modulu do pátého modulu do pravé strany. Jak by měla výsledná rekonfigurace vypadat, je vidět na obrázku z grafu z Matlabu (Obr. 68).

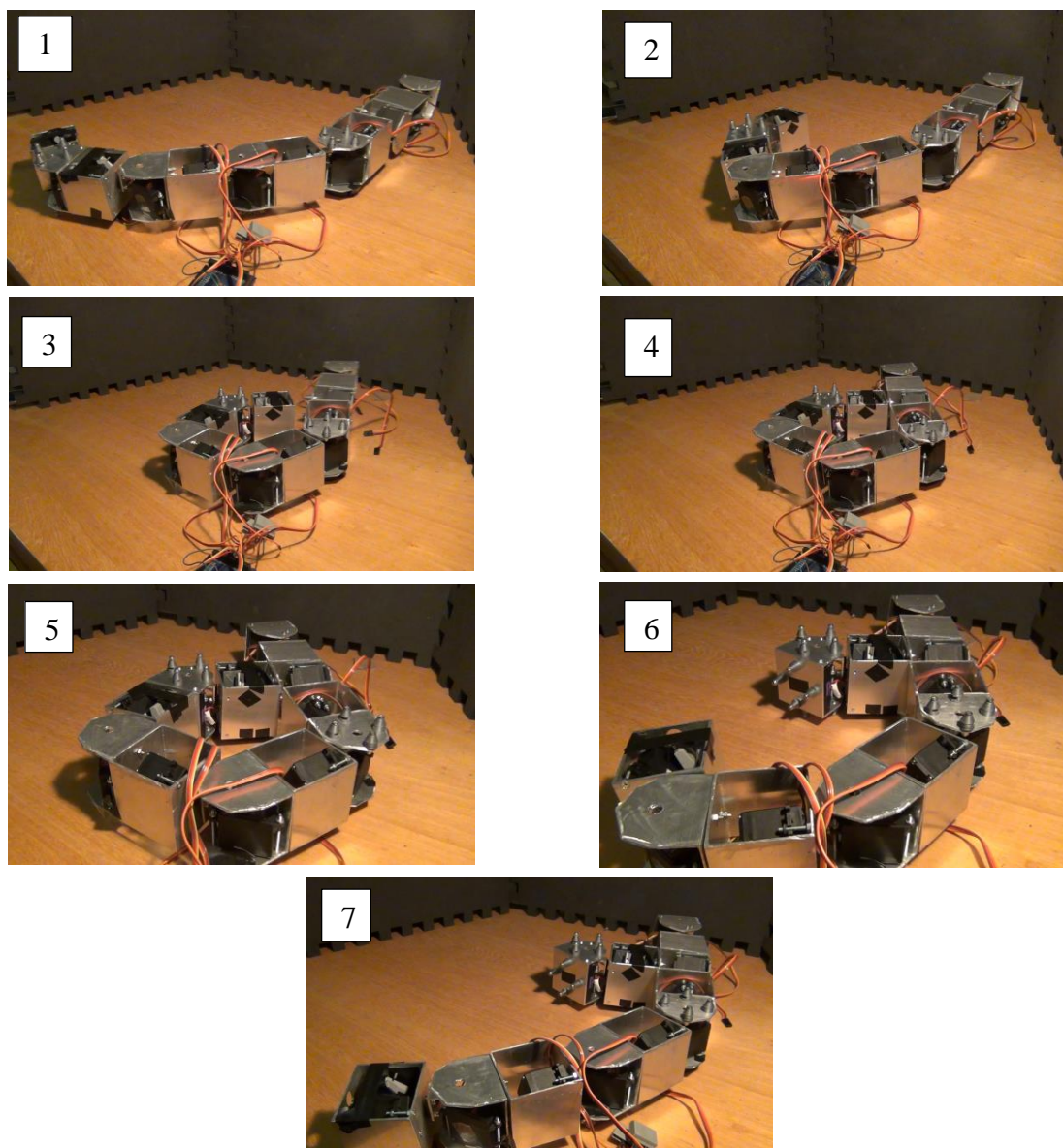


Obr. 68 Propojení s pátým modulem



Test probíhá tak, že je nejprve odesláno prvnímu modulu požadované natočení, dále druhému modulu požadované natočení, poté třetímu a nakonec čtvrtému modulu. Po ukončení pohybu čtvrtého modulu je aktivní konec již nasunutý na připojovací kužely pátého modulu. Poté se aktivní konec modulu zajistí a nově připojený modul je odpojen od sériové struktury. Odpojené moduly se od sebe oddálí a je patrný vznik nové struktury (Obr. 69). V horizontální rovině není potřebné překonávat gravitační síly, protože moduly leží na bocích a při pohybu jsou ztráty pouze třením, proto je možné realizovat všechny možnosti rekonfigurace, což je:

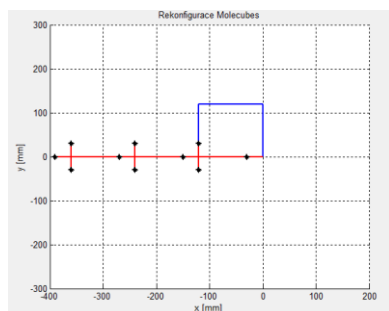
- Propojení koncového modulu se čtvrtým modulem (připojení do pravé strany, nebo do levé strany)
- Propojení koncového modulu s pátým modulem (připojení do pravé strany, nebo do levé strany)
- Propojení koncového modulu se šestým modulem (připojení do pravé strany, do levé strany, nebo zepředu)



Obr. 69 Rekonfigurace robota v horizontální rovině

### 7.14.3 Testování propojení ve vertikální rovině

V tomto testu je základní poloha robota taková, že všechny moduly jsou za sebou a úkolem je propojit koncový člen do čtvrtého modulu. Jak by měla rekonfigurace vypadat, je patrné na obrázku z Matlabu (Obr. 70) – všechny tři natočené úhly jsou rovny  $90^\circ$ .



Obr. 70 Propojení se čtvrtým modulem

Ve vertikální rovině je možná kvůli gravitačnímu působení na moduly pouze tato jediná rekonfigurace (připojení koncového členu do strany čtvrtého modulu), (Obr. 71).



*Obr. 71 Rekonfigurace robota ve vertikální rovině*

### 7.15 Poznatky z průběhu rekonfigurace

Během testování rekonfigurace bylo nutno čelit různým problémům:

1. Nepřesnost servomotorů Tower Pro MG946R

I když výrobce o nepřesnosti polohování servomotoru údaje neuvádí, tento motor má poměrně velké tření v převodech a zpětná vazba potenciometru není dostatečně citlivá, takže motor polohuje v rozmezí kolem požadovaného úhlu až s několika stupňovou chybou. To je velice nepřesné, zvláště když se jedná o

celý kinematický řetězec, který tato chyba ovlivňuje. Tyto motory mají sice velmi vysoký kroutící moment, ale pro tuto přesnou aplikaci se ukázaly jako nevhodné. Vhodnější je využít např. servomotory s označením MG995, které polohují podle provedeného měření přesně po stupních.

## 2. Proudové výkyvy

Během počátku pohybu servomotoru je odběr jednoho motoru i několik ampér. Takto vysoký odběrový proud může způsobit problémy, pokud nemáme dostatečně výkonný zdroj nebo baterii. Také vodiče musí být dostatečně dimenzované. Jinak kvůli proudovým špičkám docházelo i k výkyvům napětí a to způsobovalo resetování mikrokontroléru, který je napájený ze stejného zdroje. Vhodnými kabely a bateriemi, které byly zvoleny, se tomuto problému podařilo předejít.

## 3. Propojení vodičů

Nejprve byly vodiče uzemnění propojeny dohromady pájením, taktéž i vodiče kladného napájení, nicméně tato varianta se ukázala jako nespolehlivá a docházelo k výpadkům napájení jednotlivých motorů. Tento problém byl vyřešen použitím svorkovnice Wago, která vodiče spojuje spolehlivě.

## 4. Dotyky neizolovaných vodičů

Během testování došlo k pohybu neizolované části vodiče a došlo k dotyku s přípojevacími kolíky na desce NodeMCU. Takto byly zničeny dvě vývojové desky NodeMCU. Vyřešením byla kompletní izolace všech vodičů, tak aby nemohlo dojít k žádnému dotyku. Také byl kladen důraz na to, aby se vývojová deska nemohla dotknout stěny modulu.

## 5. Nepřesnost výroby

Kvůli nepřesné výrobě – výroba proběhla pouze ručním nářadím a s menší pečlivostí – se zjistilo, že chyby se promítají do pohybů robota, neboť osy rotace jsou mimo střed modulu. Tuto chybu zvětšila absence výkresové dokumentace k servomotorům, kdy bylo nutné si přípojevací rozměry změřit. I drobné chyby (řádově v desetinách milimetrů) ale narušují přesnost pohybů modulů. Některé moduly proto musely být opraveny, aby mohl rekonfigurační test proběhnout.

## 6. Zapnutí Molecube modulu

Při připojení baterií k modulu servomotory vždy ihned vykonaly trhavý pohyb. Teprve až se mikrokontrolér úplně spustí, nastaví se servomotory zpět do výchozí polohy. Tento pohyb je nepřijatelný, protože kdyby moduly již byly v nějaké vzájemné konfiguraci, tak by mohlo dojít ke kolizi. Proto je nutné spustit nejprve mikrokontrolér a až na něm začne běžet program (asi po 5 sekundách) připojit teprve motory. To může být realizováno buď pomocí spojovacích a rozpojovacích kolíků, mikropsínačů, nebo automaticky pomocí NPN tranzistoru.

## 7. Spojování během rekonfigurace

I když výsledky z rekonfigurace jsou úplně přesné, tak během napojování se ukázalo, že vlivem různých vůlí a nepřesností zapojení proběhne daleko lépe, když po dosednutí na přípojevací konektor udělá modul nějaký krátký natačivý pohyb, který pomůže uvolnit zaseknutí modulů bránící plnému dosednutí. Po aplikaci natačivého pohybu již dojde správně k rekonfiguraci.

## 8. Pohyb servomotorů

V počátečním nastavení se do servomotorů zasílá požadovaná hodnota natočení servomotoru a servomotor se do tohoto úhlu natočí. Nicméně servomotor se snaží pohybovat s maximální rychlostí pohybu. Celá konstrukce a hlavně převody v servomotech jsou tak vystaveny velkému silovému působení, které snižuje jejich životnost. Proto je pohyb nutné realizovat tak, aby požadovaná hodnota byla do servomotorů zasílána postupně. Toto bylo vyřešeno tím, že hodnota, kterou požadujeme je porovnávána s předchozí hodnotou a poté postupně po jednom stupni s malou časovou prodlevou (10 ms) se mezi požadovanými hodnotami natočení přechází.

## 7.16 Návrhy na zlepšení

- Pro rekonfiguraci vypočítat nejprve natočení pro bod vzdálený 20 mm kolmo před připojovaným modulem a teprve potom pro konečné dosednutí modulů. Neboť nyní je riziko, že může dojít k naražení do připojovacích kuželů při rekonfiguraci.
- Změření přesných rozměrů Molecube a systém by zpracovával přesnou polohu každého individuálního Molecube modulu v kinematickém řetězci a přepočítával by výpočet s přesnými naměřenými délkami. Tím by se zlepšila přesnost rekonfigurace.
- Implementovat navádění – pouze přiblížení k dalšímu modulu a poté přesné navádění modulů pro rekonfiguraci. Vždy by se senzory změřila chyba (odchylka vzdálenosti od středu) a přepočítal by se výpočet natočení jednotlivých kloubů.
- Ovládání rekonfigurace - nyní ovládá rekonfiguraci uživatel, ale mohl by to být expertní systém, který by podle vstupních podmínek rozhodoval o konfiguraci robota, kontroloval funkčnost modulů a kontroloval spojení modulů.

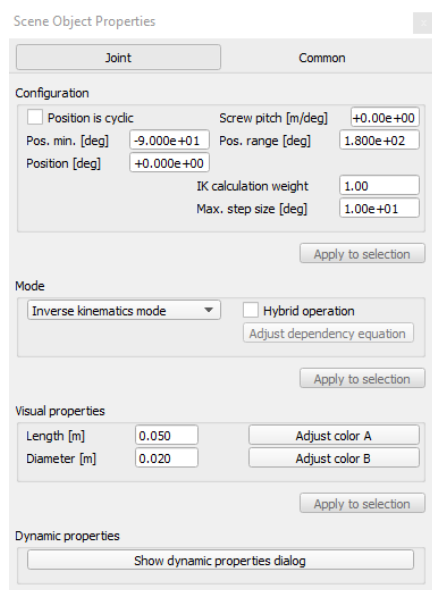
## 7.17 Manipulace ve 3D prostoru

Pro řešení složitějších úloh je možné výpočet dále rozvinout, ale ručně je to poměrně pracné. V úvahu přichází využití programu Matlab – Simulink pro vytvoření inverzní kinematiky. V tomto programu je vytvoření struktury sice poměrně jednoduché, nicméně problém omezení rozsahu rotace kloubů zde není možné jednoduše vyřešit [4].

Proto pro manipulaci ve 3D prostoru je využit program V-REP. Tento program má již zabudované ochrany proti kolizím, jednoduše nastavitelný rozsah pohybů a lze také nastavit požadovanou orientaci a polohu koncového efektoru. Jak vkládat elementy a jak je propojit a nastavit simulaci je popsáno v práci Rekonfigurace robotické struktury za použití MOLECUBES. [3]

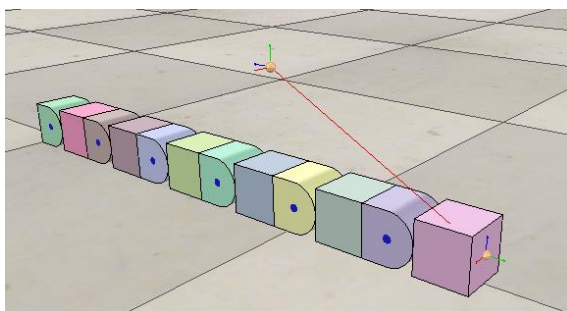
Nyní bylo nutné dodat podmínku, že je úhel rotace omezený. Tento požadavek lze nastavit ve *Scene Object Properties* kloubu (do nabídky se lze dostat označením kloubu a kliknutím na lupu na levé straně pracovního okna). V okně se zruší výběr *Position is cyclic* a nastaví se minimální hodnota na -90° a rozsah 180° (Obr. 72). Tímto je zaručený omezený rozsah pro výpočet inverzní kinematiky.



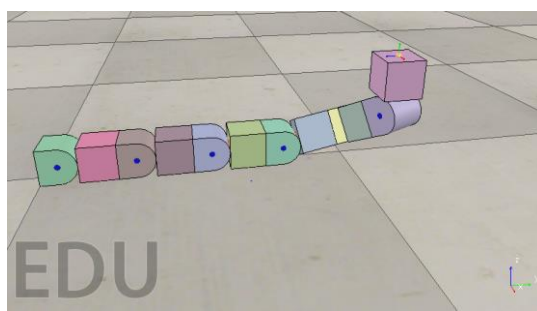


Obr. 72 Nastavení omezení rozsahu pohybu kloubu

Do programu byly nahrány zjednodušené moduly Molecube (Obr. 73) a u prvních třech modulů byl u kloubů vybrán v *Scene Object Properties* v části *Mode* místo *Inverse kinematics mode* tak *Passive mode*, aby se pro tento test moduly nenatáčely. Byl nastaven cílový bod [0,55;0,1;0,2] pro osu  $x,y,z$  v metrech a natočení [90;0;0] pro  $\alpha,\beta,\gamma$  ve stupních a byla spuštěna simulace. Z obrázku je vidět, že je robot natočený do požadované polohy (Obr. 74).



Obr. 73 Počáteční stav Molecube



Obr. 74 Molecube po simulačním výpočtu

Program V-REP má možnost vložení skriptu, který může ovládat robota nebo odesílat hodnoty natočení do reálného robota. Využití programu V-REP se tedy nabízí jako další způsob, jak ovládat Molecube, a to s využitím simulačního software.

## 7.18 Vlastnosti a využití robota Molecube

Robot tvořený moduly Molecubes je schopný se samostatně pohybovat jako had, manipulovat ve 3D prostoru a měnit samostatně svoji strukturu. Robot je vhodný pro vnitřní použití. Celková délka při šesti modulech zapojených za sebou je 72 cm. Cena komponent na jeden modul je přibližně 700 Kč, což splňuje požadavek na příznivou cenu. Celková váha robota je 2,2 kg. Díky komunikaci přes Wi-fi je možné robota ovládat jak z počítače, tak i z mobilního telefonu. Tento robot vybavený koncovými efekty (např. uchopovacím konektorem) by v budoucnu mohl být využit pro zábavní průmysl, nebo pro domácí práce.

## 8 ZÁVĚR

Po provedení analýzy problematiky sebekonfiguračních robotů byl navržen vlastní modul. Molecule modul se může natáčet ve dvou osách díky dvěma modelářským servomotorům. Nosná část modulu je kvůli snížení ceny tvořena hliníkovým profilem. Připojovací konektory jsou vytisknuty na 3D tiskárně. Každý modul má tři strany s pasivními konektory a jeden aktivní konektor. Tím je možné se připojit na připojovací konektory dalších modulů a vzájemným propojením tak vytvářet různé řetězové struktury.

Každý modul je vybavený samostatným napájením z baterií Li-ion a mikrokontrolérem, který komunikuje přes Wi-fi připojení. V programu mikrokontrolér očekává požadavky na natočení jednotlivých motorů a provádí natočení. Je také možné si ověřit, zda je připojen další modul. Úhly nutné pro rekonfiguraci jsou spočítány inverzní kinematikou metodou nejmenších čtverců. Výpočet je realizován podle iteračního algoritmu v programu Matlab a výpočet je optimalizovaný pro rekonfiguraci tak, aby byl dodržen maximální rozsah pohybů kloubů robota.

Posledním cílem bylo vytvořit jednoduché struktury. Proto bylo vyrobeno 6 modulů, na kterých bylo testováno provedení změny struktury. Ta spočívala v tom, že ze sériového připojení modulů za sebou se struktura změní tak, aby byl koncový modul připojený do strany jiného modulu. Rekonfigurace proběhla úspěšně a ověřila se tím funkčnost programu pro mikrokontrolér, připojovacích konektorů, volba napájení a výpočet inverzní kinematiky.





## 9 SEZNAM POUŽITÝCH ZDROJŮ

- [1] Modular Self-Reconfigurable Robots. YIM, Mark, Paul WHITE, Michael PARK a Jimmy SASTRA. *Encyclopedia of Complexity and System Science* [online]. 2009, s. 19-31 [cit. 2016-05-13]. ISBN 978-0-387-75888-6. Dostupné z: [http://www.springer.com/cda/content/document/cda\\_downloaddocument/9780387758886-c2.pdf?SGWID=0-0-45-621405-p173779107](http://www.springer.com/cda/content/document/cda_downloaddocument/9780387758886-c2.pdf?SGWID=0-0-45-621405-p173779107)
- [2] STOY, Kasper, David BRANDT a David J. CHRISTENSEN. *Self-reconfigurable robots* [online]. 2010 [cit. 2016-05-14]. ISBN 978-0-262-01371-0. Dostupné z: [https://mitpress.mit.edu/sites/default/files/titles/content/9780262013710\\_sch\\_0001.pdf](https://mitpress.mit.edu/sites/default/files/titles/content/9780262013710_sch_0001.pdf)
- [3] VÍTEK, Filip. *Konfigurace robotické struktury za využití MOLECUBES*. Brno, 2015. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Doc. Ing. Simeon Simeonov, CSc.
- [4] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Brno: CERM, 2007. ISBN 978-80-214-3530-8.
- [5] M-TRAN III. *AIST* [online]. 2010 [cit. 2016-05-14]. Dostupné z: <https://unit.aist.go.jp/is/frrg/dsysd/mtran3/mtran3.htm>
- [6] GILPIN, Kyle, Keith KOTAY a Daniela RUS. *Miche: Modular Shape Formation by Self-Dissassembly* [online]. [cit. 2016-05-14]. Dostupné z: <https://groups.csail.mit.edu/drl/wiki/images/b/b5/Miche-GilpinKotayRus-ICRA07.pdf>
- [7] DOCTORADO, Juan. *Generación G2* [online]. 2003 [cit. 2016-05-14]. Dostupné z: <http://www.learobotics.com/personal/juan/doctorado/tea/html/node42.html>
- [8] *Swarm-bots: Swarm-bots hardware* [online]. 2005 [cit. 2016-05-14]. Dostupné z: <http://www.swarm-bots.org/index.php@main=3&sub=31&conpage=sbot.html>
- [9] SENGUPTA, Rustam a Ankit GUPTA. *Modular Robots- Enhancement in Robotic Technology by the development of Segmented Reconfigurable High-Utility Robots* [online]. [cit. 2016-05-14]. Dostupné z: [http://xenon.stanford.edu/~caesars/rustams/Modular\\_Robotics](http://xenon.stanford.edu/~caesars/rustams/Modular_Robotics)
- [10] A Chain Reconfiguration Robot. : *Generation III* [online]. 2007 [cit. 2016-05-14]. Dostupné z: <http://elek3ronik.blogspot.cz/2007/05/chain-reconfiguration-robot.html>
- [11] ØSTERGAARD, Esben Hallundbæk, Kristian KASSOW, Richard BECK a Henrik Hautop LUND.

- Design of the ATRON lattice-based self-reconfigurable robot. *Auton Robot* [online]. 2006, , 165-183 [cit. 2016-05-14]. DOI: 10.1007/s10514-006-8546-1. Dostupné z: <http://www.itu.dk/~ksty/uploads/publications/ostergaard2006ar.pdf>
- [12] ZYKOV, Viktor. *MORPHOLOGICAL AND BEHAVIORAL RESILIENCE AGAINST PHYSICAL DAMAGE FOR ROBOTIC SYSTEMS*[online]. Cornell University, 2008 [cit. 2016-05-14]. Dostupné z: <http://creativemachines.cornell.edu/papers/ZykovPhDThesis.pdf>
- [13] Robot, nebo had? CAD [online]. [cit. 2016-05-14]. Dostupné z: <http://www.cad.cz/strojirenstvi/38-strojirenstvi/1914-robot-nebo-had.html>
- [14] STØY, K., W.-M. SHEN a P. WILL. *How to Make a Self-Reconfigurable Robot Run* [online]. 2002 [cit. 2016-05-14]. Dostupné z: [https://groups.csail.mit.edu/mac/projects/amorphous/6.978/papers/stoy\\_aamas02.pdf](https://groups.csail.mit.edu/mac/projects/amorphous/6.978/papers/stoy_aamas02.pdf)
- [15] MAYNARD, James. Roombot modular robot can turn into any piece of furniture. In: *Techtimes* [online]. 2014 [cit. 2016-05-14]. Dostupné z: <http://www.techtimes.com/articles/7575/20140527/roombot-modular-robot-can-turn-into-any-piece-of-furniture-autobots-you-have-competition.htm>
- [16] Arduino - Introduction. *Arduino* [online]. 2016 [cit. 2016-05-14]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [17] Arduino Pro Mini. *Arduino* [online]. [cit. 2016-05-14]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardProMini>
- [18] TowerPro MG946R 12kg / 0.20sec / 55g. In: *Hobbyking* [online]. [cit. 2016-05-21]. Dostupné z: [http://www.hobbyking.com/hobbyking/store/\\_6222\\_TowerPro\\_MG946R\\_12kg\\_0\\_20sec\\_55g.html](http://www.hobbyking.com/hobbyking/store/_6222_TowerPro_MG946R_12kg_0_20sec_55g.html)
- [19] SG90. In: *Micropik* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.micropik.com/PDF/SG90Servo.pdf>
- [20] 2\*AA Parallel Batteries Holder Case Box with Leads. In: *FastTech* [online]. [cit. 2016-05-14]. Dostupné z: <https://www.fasttech.com/product/1289504-2aa-parallel-batteries-holder-case-box-with-leads>
- [21] Komunikujeme bezdrátově s NRF24L01 – 1. část – zapojení. In: *Arduino8* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.arduino8.cz/komunikujeme-bezdratove-s-nrf24l01-1-cast-zapojeni/>

- [22] DC-DC Adjustable Step-up Boost Power Converter Module XL6009 Better than LM2577. *ShopClues* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.shopclues.com/dc-dc-adjustable-step-up-boost-power-converter-module-xl6009-better-than-lm2577.html>
- [23] Jekl 60 x 60 x 2,0 mm /6bm. In: *Aluminiumshop* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.aluminiumshop.cz/?i=628/jekl-60-x-60-x-2-0-mm-6bm&s=133#>
- [24] YIM, MARK, WEI-MIN SHEN a BEHNAM SALEMI. *Modular Self-Reconfigurable Robot Systems* [online]. 2007, , 43-52 [cit. 2016-05-14]. Dostupné z: [http://creativemachines.cornell.edu/papers/RAM07\\_Yim.pdf](http://creativemachines.cornell.edu/papers/RAM07_Yim.pdf)
- [25] SABATINI, MATTHEW. Lithium Ion vs. Lithium Polymer – What’s the Difference? In: *Androidauthority* [online]. 2011 [cit. 2016-05-14]. Dostupné z: <http://www.androidauthority.com/lithium-ion-vs-lithium-polymer-whats-the-difference-27608>
- [26] 3V-12V DC 80mA-350mA Micro solenoid electromagnet push and pull dc Miniture electromagnet Mini Solenoid Electromagnet. In: *Amazon* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.amazon.com/80mA-350mA-solenoid-electromagnet-Miniture-Electromagnet/dp/B00ZC53KB4>
- [27] NodeMCU. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2016-05-14]. Dostupné z: <https://en.wikipedia.org/wiki/NodeMCU>
- [28] NodeMCU WiFi modul s ESP8266. In: *Arduinotech* [online]. [cit. 2016-05-14]. Dostupné z: <https://www.arduinotech.cz/inpage/nodemcu-wifi-modul-s-esp8266>
- [29] NodeMCU v2 - Lua based ESP8266 development kit. In: *Seedstudio* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.seedstudio.com/depot/NodeMCU-v2-Lua-based-ESP8266-development-kit-p-2415.html>
- [30] Mikrospínač ZIPPY SM-05H-07P0-Z. In: *Gme* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.gme.cz/mikrospinac-zippy-sm-05h-07p0-z-p630-037>
- [31] NodeMCU is both a Breadboard-Friendly ESP8266 Wi-Fi Board and a LUA based Firmware. In: *Cnx-software* [online]. 2015 [cit. 2016-05-14]. Dostupné z: <http://www.cnx-software.com/2015/04/18/nodemcu-is-both-a-breadboard-friendly-esp8266-wi-fi-board-and-a-lua-based-firmware/>
- [32] GTF 4pcs 3.7V 14500 2300mAh Li-ion Rechargeable Battery For Flashlight Torch. In: *Amazon* [online]. [cit. 2016-05-14]. Dostupné z: [http://www.amazon.com/2300mAh-Li-ion-Rechargeable-Battery-Flashlight/dp/B019GKAQB2/ref=sr\\_1\\_2?s=hi&ie=UTF8&qid=1463247114&sr=1-2&keywords=14500+2300](http://www.amazon.com/2300mAh-Li-ion-Rechargeable-Battery-Flashlight/dp/B019GKAQB2/ref=sr_1_2?s=hi&ie=UTF8&qid=1463247114&sr=1-2&keywords=14500+2300)

- [33] 5in1 USB Charger + 5pcs 3.7V 25C 850mAh Li-Po Battery for Syma X5SW X5SC RC176. In: *Ebay* [online]. [cit. 2016-05-14]. Dostupné z: [http://www.ebay.com/itm/5in1-USB-Charger-5pcs-3-7V-25C-850mAh-Li-Po-Battery-for-Syma-X5SW-X5SC-RC176-/172103560415?hash=item28122becdf%3A%3A~AAASwezVWwq6j&nma=true&si=VvaEXNfcm01sKb6naHdH4unPIBs%253D&orig\\_cvip=true&rt=nc&trksid=p2047675.l2557](http://www.ebay.com/itm/5in1-USB-Charger-5pcs-3-7V-25C-850mAh-Li-Po-Battery-for-Syma-X5SW-X5SC-RC176-/172103560415?hash=item28122becdf%3A%3A~AAASwezVWwq6j&nma=true&si=VvaEXNfcm01sKb6naHdH4unPIBs%253D&orig_cvip=true&rt=nc&trksid=p2047675.l2557)
  
- [34] USB – Universal Serial Bus – Univerzální Sériová Sběrnice. In: *Z-moravec* [online]. [cit. 2016-05-14]. Dostupné z: <http://z-moravec.net/elektronika/vypocetni-technika/usb-universal-serial-bus-univerzalni-seriova-sbernice/>
  
- [35] Servo attach. In: *Arduino* [online]. [cit. 2016-05-14]. Dostupné z: <https://www.arduino.cc/en/Reference/ServoAttach>
  
- [36] New MacBook is a slim, light and eye-catching notebook. In: *Usatoday* [online]. [cit. 2016-05-14]. Dostupné z: <http://www.usatoday.com/story/tech/2015/03/09/apple-new-macbook/24661201/>
  
- [37] TP-Link TL-WR841ND 300Mbps Wireless LAN Router. In: *Ekatalog* [online]. [cit. 2016-05-14]. Dostupné z: [http://ekatalog.ispalliance.cz/cz/produkt/tp-link-tl-wr841nd-300mbps-wireless-lan-router\\_1/](http://ekatalog.ispalliance.cz/cz/produkt/tp-link-tl-wr841nd-300mbps-wireless-lan-router_1/)
  
- [38] REED, Frances. How Do Servo Motors Work. In: *Jameco* [online]. c2012-2016 [cit. 2016-05-14]. Dostupné z: <http://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>
  
- [39] HTTPServo. In: *GitHub* [online]. 2015 [cit. 2016-05-14]. Dostupné z: <https://gist.github.com/igrr/9ef4d5c74355503e3b1f>
  
- [40] HelloServer. In: *GitHub* [online]. 2015 [cit. 2016-05-14]. Dostupné z: <https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WebServer/examples/HelloServer/HelloServer.ino>
  
- [41] WiFiAccessPoint. In: *GitHub* [online]. [cit. 2016-05-14]. Dostupné z: <https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/examples/WiFiAccessPoint/WiFiAccessPoint.ino>
  
- [42] MACICH, Jiří. Jak na domácí síť 1: základní infrastruktura. In: *Pc-help* [online]. 2013 [cit. 2016-05-14]. Dostupné z: <http://www.pc-help.cz/viewtopic.php?t=104106>

# 10 SEZNAM OBRÁZKŮ A SYMBOLŮ

## 10.1 Seznam obrázků

Obr. 1 Využití ve vesmíru [24] .....	18
Obr. 2 Robotický nábytek Roombot [15] .....	18
Obr. 3 Robot Míche [6] .....	18
Obr. 4 Polybot [7] .....	18
Obr. 5 Swarm-bot mobilní robot [8] .....	19
Obr. 6 M-TRAN III [1] .....	19
Obr. 7 Přeměna kráčejičího robota na tvar hada [5] .....	22
Obr. 8 Vzhled M-TRAN modulu [5] .....	22
Obr. 9 Připojovací plocha modulu [5] .....	22
Obr. 10 Detail propojení jednotlivých modulů [5] .....	22
Obr. 11 Polybot G3 [10] .....	23
Obr. 12 Atron robot [11] .....	24
Obr. 13 Atron vnitřek modulu [11] .....	24
Obr. 14 Symetrické tvary modulů [12] .....	25
Obr. 15 Kinematické uspořádání robota [13] .....	26
Obr. 16 Fotografie robota hada [13] .....	26
Obr. 17 Konstrukční návrh (varianta a) .....	26
Obr. 18 Konstrukční návrh (varianta b) .....	26
Obr. 19 Připojovací kužely .....	27
Obr. 20 Připojovací kříž .....	27
Obr. 21 Elektromagnet [26] .....	27
Obr. 22 Servomotor Tower Pro SG90 [19] .....	27
Obr. 23 Připojovací kužel .....	28
Obr. 24 Servo Tower Pro [18] .....	29
Obr. 25 Silové působení na moduly .....	30
Obr. 26 Zkrácené páky servomotoru .....	31
Obr. 27 Modul po 3D vytisknutí .....	31
Obr. 28 Hliníkový profil [23] .....	31
Obr. 29 Mikrospínač [30] .....	32
Obr. 30 Uchycení kontrolního mikrospínače .....	32
Obr. 31 Umístění vstupů a výstupů na Vývojové desce [31] .....	33
Obr. 32 Modul v krajní poloze a maximálně natočen .....	34
Obr. 33 Umístění otvoru pro průchod kabelu .....	34
Obr. 34 Celkový pohled na modul .....	34
Obr. 35 Mikrokontrolér Arduino Pro Mini [17] .....	35
Obr. 36 Bezdrátový modul NRF24L01 [21] .....	36
Obr. 37 Vývojová deska NodeMCU [29] .....	37
Obr. 38 Li-ion baterie 14500 [32] .....	38
Obr. 39 Li-po baterie [33] .....	38
Obr. 40 Držák baterie AA [20] .....	39

Obr. 41 Stabilizátor napětí DSN6009 [22] .....	39
Obr. 42 USB schéma [34] .....	40
Obr. 43 Vývojové prostředí .....	41
Obr. 44 Výběr sériového portu .....	42
Obr. 45 Mód s vytvořením přístupového bodu [36] .....	43
Obr. 46 Mód s využitím routeru [36][37] .....	43
Obr. 47 Komponenty servomotoru [38] .....	43
Obr. 48 Délka pulzu ovlivňuje natočení servomotoru [38] .....	43
Obr. 49 Pro propojení dvou modulů musíme znát úhly natočení .....	49
Obr. 50 Otevřený kinematický řetězec [4] .....	49
Obr. 51 Rotace dvou souřadnicových systémů [4] .....	50
Obr. 52 Posun a natočení dvou souřadnicových systému v rovině [4] .....	51
Obr. 53 Manipulace mezi překážkami .....	54
Obr. 54 Hodnoty parametru $\lambda$ jsou stejné .....	54
Obr. 55 Úprava hodnot $\lambda$ tak, aby záleželo více na cílovém bodě .....	54
Obr. 56 Manipulace za překážkou .....	54
Obr. 57 Natočení propojovaného modulu .....	55
Obr. 58 Požadovaná rekonfigurace .....	59
Obr. 59 Nevhodné řešení rekonfigurace .....	59
Obr. 60 Rekonfigurace po omezení natočení v kloubech .....	60
Obr. 61 Jednotlivá řešení během výpočtu .....	60
Obr. 62 Neschopnost nelézt přesné řešení .....	61
Obr. 63 Konvergence nastane ihned po překročení 500 iterací .....	61
Obr. 64 Úhel natočení koncového efektoru .....	62
Obr. 65 Konvergence po upravených poč. podmínkách .....	62
Obr. 66 Okamžité nalezení řešení .....	62
Obr. 67 Grafické uživatelské rozhraní .....	63
Obr. 68 Propojení s pátým modulem .....	64
Obr. 69 Rekonfigurace robota v horizontální rovině .....	66
Obr. 70 Propojení se čtvrtým modulem .....	66
Obr. 71 Rekonfigurace robota ve vertikální rovině .....	67
Obr. 72 Nastavení omezení rozsahu pohybu kloubu .....	70
Obr. 73 Počáteční stav Molecube .....	70
Obr. 74 Molecube po simulačním výpočtu .....	70

## 10.2 Seznam symbolů

$F_g$	Tíhová síla [N]
$r$	Poloměr [m]
$M_k$	Kroutící moment [Nm]
$I$	Moment setrvačnosti tělesa [ $kg \cdot m^2$ ]
$\varepsilon$	Úhlové zrychlení [ $rad/s^2$ ]
$m$	Hmotnost [kg]
$g$	Gravitační zrychlení [ $m/s^2$ ]
<b>J</b>	Jakobián, Jacobiho matice
$\varphi$	Úhel natočení [rad]
<b>R</b>	Rotační matice
$R_{ba}$	Rotační matice vyjadřující natočení systému $b$ vůči $a$ vyjádřená v systému $a$
$r_b^M$	Poloha bodu $M$ v systému $b$
<b>q</b>	Vektor zobecněných kloubových souřadnic
$\Delta x$	Vektor rozdílů mezi požadovanou a skutečnou hodnotou kartézských souřadnic
$\tilde{x}$	Vektor požadovaných hodnot kartézských souřadnic
$J^T$	Transpozice Jakobiánu
$J^+$	Pseudoinverze Jakobiánu
$\lambda$	Tlumičí konstanta
$R_{ij}$	Rozšířená rotační matice (dvojnásobná)





## 11 SEZNAM PŘÍLOH

Příloha 1    CD obsahující:

- Elektronickou verzi diplomové práce (pdf)
- Výrobní výkres: Horní profil (pdf)
- Výrobní výkres: Spodní profil (pdf)
- Model modulu Molecube (ipt)
- Program pro Matlab (m)
- Program pro mikrokontrolér (ino)
- Simulace V-REP (ttt)